



INSTITUTO POLITÉCNICO  
DE VIANA DO CASTELO

Gilberto Gonalo Gomes Ribeiro  
Software para  
Gesto de Projetos de Software

Mestrado em Tecnologias e Gesto de Sistemas de Informao

Trabalho de Projeto efetuado sob a orientao do  
Doutor Antnio Miguel Cruz

Novembro de 2015



## RESUMO

A gestão de projetos de software dá suporte a todas atividades que visam assegurar que o sistema ou software seja entregue no prazo pré-definido e esteja de acordo com os requisitos definidos pelo cliente.

Essa necessidade de gestão de projetos deve-se exatamente ao facto do desenvolvimento de software estar sempre sujeito às restrições de qualidade, tempo e custos, dos métodos e das tarefas a seguir, bem como das ferramentas a utilizar em cada fase.

Por todos estes fatores e pela evolução tecnológica, o interesse nas soluções Open Source tornou-se inquestionavelmente global.

Os sistemas de gestão de projetos de software sustentados por tecnologia Open Source não são abundantes, e ainda continuam a ser restritivos em relação às soluções comerciais, ou de “mercado fechado”.

Baseado na crescente necessidade de haver alternativas abertas com inovação tecnológica no processo de gestão de projetos de software, este trabalho pretende analisar, modelar e desenvolver uma ferramenta que dê suporte digital contínuo a todas as etapas da gestão de projetos de software, e que ajude a fazer a ponte entre tarefas de análise e modelação e tarefas de gestão com elas relacionadas.

Neste projeto foi desenvolvida uma aplicação web java, baseada em Spring MVC para gestão de desenvolvimento de software.

As funcionalidades de maior relevo incluem a existência de uma gestão de utilizadores que escalona graus de tarefas diferentes para diferentes tarefas, o acompanhamento em tempo real de todos os utilizadores em cada projeto, o mapeamento de todas as tarefas e elementos de modelos de software nos requisitos do projeto.

Novembro de 2015



## ABSTRACT

Project management software supports all activities aimed at ensuring that the system or software is delivered in pre-set time and complies with the requirements set by the client.

This need for project management comes exactly from the software development process, which is always subject to quality constraints, time and costs, the methods and tasks to follow, and the tools to use in each phase.

For all these factors and technological developments, interest in Open Source solutions has become undeniably global.

Software project management systems supported by Open Source technology are not abundant and are still more restrictive than commercial solutions, or those from “closed market”.

Based on the growing need for alternatives open to technological innovation in the software project management process, this Master of Science work aims to analyze, model and develop a tool that gives continuous digital support for all stages of software project management, and that helps bridging the gap between analysis and modeling tasks and management tasks related to them.

In this project, a java web application has been developed based on Spring MVC for software development management.

The most prominent features include the existence of a user management which scales degrees of different tasks for different tasks, the real-time monitoring of all users on each project, the mapping of all tasks and software model elements to requirements from the project.

November 2015



# CONTEÚDO

1. Introdução .....	1
1.1 A gestão de Projetos de Software .....	2
1.2 A Tecnologia Open Source .....	5
1.3 Apresentação de objetivos do Projeto .....	7
1.4 Estrutura da Dissertação .....	9
2. Definições e Conceitos .....	11
2.1 Introdução .....	11
2.2 A arquitetura MVC .....	12
2.3 O java persistence api e framework orm .....	14
2.4 A framework spring mvc .....	15
2.5 Tecnologias e conceitos utilizados .....	16
2.5.1 Postgresql 9.3 .....	16
2.5.2 StarUml .....	17
2.5.3 Balsamiq Mockups .....	17
2.5.4 Spring Tool Suite .....	18
2.5.5 Thymeleaf .....	20
2.5.6 Foundation Template .....	20
2.5.7 JQuery .....	21
2.6 Notas Finais .....	21
3. Estudo do Estado da Arte .....	23
3.1 Introdução .....	23
3.2 Abordagens Existentes .....	24
3.3 A importância do conhecimento nas aplicações de gestão de projetos .....	29
3.4 O PMBOK e a importância do conhecimento na gestão de projetos .....	30
3.4.1 Ciclo de vida de um projeto .....	33
3.4.2 Áreas de conhecimento na Gestão de Projetos .....	35
3.4.3 Os 47 processos do PMBOK .....	39
3.4.4 A estrutura de gestão de Projetos (PMO) .....	41
3.4.5 A Certificação Project Management Professional(PMP) .....	42
3.4.6 Gestor de Projeto e as partes interessadas do projeto .....	43

3.4.7 Fatores críticos de sucesso e insucesso na gestão de projetos .....	46
3.5 Conclusões ou Notas Finais .....	47
4. Análise do Problema .....	51
4.1 Introdução .....	51
4.2 Requisitos .....	52
4.3 Estruturação dos Requisitos.....	57
4.4 Casos de Uso .....	67
4.5 Modelo de Domínio.....	82
4.6 Storyboards do Sistema .....	88
4.7 Tarefas automáticas do Sistema .....	103
4.8 Conclusões ou notas finais .....	106
5. Desenvolvimento do Projeto .....	111
5.1 Introdução .....	111
5.2 Desenho da aplicação.....	111
5.3 Comportamento.....	114
5.4 Conclusões ou Notas Finais .....	121
6. Conclusões e Trabalho Futuro .....	123
6.1 Introdução .....	123
6.2 Contributo .....	124
6.3 Discussão de resultados .....	124
6.4 Principais Resultados Obtidos .....	126
6.5 Trabalho Futuro.....	128
Referências.....	131
Referências WWW .....	132
Anexos.....	135
Anexo A: Storyboards.....	135
A.1 Storyboard do Project Manager .....	135
A.2 Storyboard do Team Member .....	149
A.3 Storyboard do Stakeholder.....	160
A.4 Storyboard do Administrador.....	165
Anexo B: Screens da aplicação .....	170



B.1 Screens do Administrador .....	170
B.2 Screens do Project Manager .....	172



# ÍNDICE DE FIGURAS

FIGURA 1 - ARQUITETURA MVC (FREEMN & FREEMAN P. 424).....	12
FIGURA 2 - OUTRO ESQUEMA DA ARQUITETURA MVC (RETIRADO DE [ARQUITETURA MVC]).....	13
FIGURA 3 - ARQUITETURA SPRING MVC (SPRING SCAFFOLDING SUPPORT IN MYECLIPSE).....	15
FIGURA 4 – AMBIENTE GRÁFICO DO BALSAMIQ MOCKUPS.....	18
FIGURA 5 - FERRAMENTA DE GESTÃO DE PROJETOS DA OPEN PROJECT FOUNDATION.....	26
FIGURA 6 - FERRAMENTA WRIKE PARA GESTÃO DE PROJETOS.....	27
FIGURA 7 - FERRAMENTA AGILEWRAP PARA GESTÃO DE PROJETOS.....	29
FIGURA 8 - PORTFÓLIOS, PROGRAMAS E PROJETOS (RETIRADO DE PMBOK GUIDE, 2013).....	32
FIGURA 9 - GRUPOS DE PROCESSOS DE GESTÃO DE PROJETOS (RETIRADO DE BARBOSE FERNANDO, GERENCIAMENTO DE PROJETOS).....	33
FIGURA 10 - ÁREAS DE CONHECIMENTO DO PMBOK (RETIRADO DE PMBOK GUIDE, 2013).....	35
FIGURA 11 - RELAÇÃO ENTRE A EQUIPA DE PROJETO E OUTROS STAKEHOLDERS (RETIRADO DE PMBOK GUIDE, 2013).....	46
FIGURA 12 - DIAGRAMA DE PACOTES DA APLICAÇÃO.....	58
FIGURA 13 - COMPOSIÇÃO DO PACOTE DE GESTÃO DE EQUIPA.....	63
FIGURA 14 - COMPOSIÇÃO DO PACOTE DE GESTÃO DE PROJETO.....	66
FIGURA 15 - DIAGRAMA DE CASOS DE USO DO PROJECT MANAGER.....	70
FIGURA 16 - DIAGRAMA DE CASOS DE USO DO TEAM MEMBER.....	75
FIGURA 17 - DIAGRAMA DE CASOS DE USO DO STAKEHOLDER PROPONENTE.....	78
FIGURA 18 - DIAGRAMA DE CASOS DE USO DO ADMINISTRADOR DO SISTEMA.....	81
FIGURA 19 - DIAGRAMA DE CLASSES DO DOMÍNIO DO PROJETO.....	84
FIGURA 20 - STORYBOARD DO LOGIN DA APLICAÇÃO.....	89
FIGURA 21 - OPERAÇÕES ADMINISTRATIVAS DO PROJETO.....	90
FIGURA 22 - OPERAÇÕES COMUNS DO PROJETO.....	91
FIGURA 23 - COMPOSIÇÃO DO PACOTE DE GESTÃO DE HISTÓRICOS.....	104
FIGURA 24 - CRONOGRAMA DO PROJETO.....	108
FIGURA 25 - CRONOGRAMA DE TAREFAS POR UTILIZADOR DO PROJETO.....	109
FIGURA 26 - DECOMPOSIÇÃO ESTRUTURAL DE APLICAÇÃO.....	113
FIGURA 27 - ADMINISTRATOR - LOGIN DA APLICAÇÃO.....	115
FIGURA 28 - ADMINISTRATOR - OPERAÇÕES ADMINISTRATIVAS.....	116
FIGURA 29 - ADMINISTRATOR - OPERAÇÕES DO PROJECT MANAGER.....	118
FIGURA 30 - TEAM MEMBER – OPERAÇÕES DO TEAM MEMBER.....	119
FIGURA 31 - STAKEHOLDER – OPERAÇÕES DO STAKEHOLDER.....	120
FIGURA 32 - PROJECT MANAGER – LOGIN.....	135
FIGURA 33 - PROJECT MANAGER - MEUS PROJETOS.....	136
FIGURA 34 - PROJECT MANAGER - LISTA DOS TEAM MEMBERS.....	136
FIGURA 35 - PROJECT MANAGER - TEAM MEMBER.....	137
FIGURA 36 - PROJECT MANAGER - LISTA DE STAKEHOLDERS.....	137
FIGURA 37 - PROJECT MANAGER - STAKEHOLDER.....	138
FIGURA 38 - PROJECT MANAGER - EQUIPA DE ANÁLISE.....	138
FIGURA 39 - PROJECT MANAGER - LISTA DE REQUISITOS.....	139
FIGURA 40 - PROJECT MANAGER - REQUISITO.....	140
FIGURA 41 - PROJECT MANAGER - LISTA DE CASOS DE USO.....	141
FIGURA 42 - PROJECT MANAGER - CASO DE USO.....	142

FIGURA 43 - PROJECT MANAGER - LISTA DE TAREFAS .....	143
FIGURA 44 - PROJECT MANAGER - TAREFA .....	144
FIGURA 45 - PROJECT MANAGER - ASSOCIAÇÃO DE REQUISITOS.....	145
FIGURA 46 - PROJECT MANAGER - ASSOCIAÇÃO DE CASOS DE USO. ....	145
FIGURA 47 - PROJECT MANAGER - ASSOCIAÇÃO DE TAREFAS.....	146
FIGURA 48 - PROJECT MANAGER - LISTA DE PEDIDOS DE ALTERAÇÃO.....	146
FIGURA 49 - PROJECT MANAGER – PEDIDO DE ALTERAÇÃO .....	147
FIGURA 50 - PROJECT MANAGER - LISTA DE VOTAÇÕES.....	147
FIGURA 51 - PROJECT MANAGER – VOTO. ....	148
FIGURA 52 - TEAM MEMBER - LOGIN.....	149
FIGURA 53 - TEAM MEMBER - MEUS PROJETOS. ....	149
FIGURA 54 - TEAM MEMBER - PROJETO. ....	150
FIGURA 55 - TEAM MEMBER - LISTA DE REQUISITOS.....	150
FIGURA 56 - TEAM MEMBER - REQUISITO. ....	151
FIGURA 57 - TEAM MEMBER - LISTA DE CASOS DE USO. ....	152
FIGURA 58 - TEAM MEMBER - CASO DE USO. ....	153
FIGURA 59 - TEAM MEMBER - LISTA DE TAREFAS.....	154
FIGURA 60 - TEAM MEMBER - TAREFA.....	155
FIGURA 61 - TEAM MEMBER - ASSOCIAÇÃO DE REQUISITOS. ....	156
FIGURA 62 - TEAM MEMBER - ASSOCIAÇÃO DE CASOS DE USO. ....	156
FIGURA 63 - TEAM MEMBER - LISTA DE TAREFAS.....	157
FIGURA 64 - TEAM MEMBER - LISTA DE PEDIDOS DE ALTERAÇÃO. ....	157
FIGURA 65 - TEAM MEMBER - PEDIDO DE ALTERAÇÃO.....	158
FIGURA 66 - TEAM MEMBER - LISTA DE VOTAÇÕES.....	158
FIGURA 67 - TEAM MEMBER – VOTAÇÃO. ....	159
FIGURA 68 - STAKEHOLDER - LOGIN.....	160
FIGURA 69 - STAKEHOLDER - MEUS PROJETOS. ....	160
FIGURA 70 - STAKEHOLDER - PROJETO.....	161
FIGURA 71 - LISTA DE REQUISITOS. ....	161
FIGURA 72 - STAKEHOLDER - REQUISITO. ....	162
FIGURA 73 - STAKEHOLDER - LISTA DE PEDIDOS DE ALTERAÇÃO. ....	162
FIGURA 74 - STAKEHOLDER - PEDIDO DE ALTERAÇÃO. ....	163
FIGURA 75 - STAKEHOLDER - LISTA DE VOTAÇÕES.....	163
FIGURA 76 - : STAKEHOLDER - VOTAÇÃO. ....	164
FIGURA 77 - ADMINISTRATOR - LOGIN.....	165
FIGURA 78 - ADMINISTRATOR - LISTA DE UTILIZADORES.....	165
FIGURA 79 - ADMINISTRATOR - UTILIZADOR. ....	166
FIGURA 80 - ADMINISTRATOR - LISTA DE HABILIDADES.....	166
FIGURA 81 - ADMINISTRATOR - HABILIDADE. ....	167
FIGURA 82 - ADMINISTRATOR - MEUS PROJETOS.....	167
FIGURA 83 - ADMINISTRATOR – PROJETO.....	168
FIGURA 84 - ADMINISTRATOR - OUTROS PROJETOS. ....	168
FIGURA 85 - ADMINISTRATOR - OUTRO PROJETO.....	169
FIGURA 86 - ADMINITRATOR - SCREEN DO UTILIZADOR.....	170
FIGURA 87 - ADMINISTRATOR - SCREEN DA HABILIDADE .....	171
FIGURA 88 - ADMINISTRATOR - SCREEN DA MUDANÇA DO CHEFE DE PROJETO. ....	171
FIGURA 89 - PROJECT MANAGER - ASSOCIAR MEMBRO DE EQUIPA. ....	172

FIGURA 90 - : PROJECT MANAGER - ASSOCIAR STAHEHOLDER.....	173
FIGURA 91 - PROJECT MANAGER - EQUIPA DE ANÁLISE. ....	174



## ÍNDICE DE TABELAS

TABELA 1 - GRUPOS DE PROCESSOS E ÁREAS DE CONHECIMENTO DO PROJETO (RETIRADO DA REVISTA MUNDOPM, 2015). .....	39
TABELA 2 - TABELA COMPARATIVA ENTRE O OPEN PROJECT FOUNDATION E A APLICAÇÃO DESENVOLVIDA. ....	125





## AGRADECIMENTOS

Gostaria de expressar os meus sinceros agradecimentos a todos aqueles que de forma direta ou indireta contribuíram para a realização deste trabalho.

Em primeiro lugar, gostaria de agradecer aos meus pais e familiares, que tão carinhosamente me apoiaram ao longo deste mestrado. Estiveram sempre presentes, não deixaram nunca de me apoiar nos momentos mais difíceis bem como pela imensa força, apoio e compreensão disposta.

Ao Prof. Doutor António Miguel Cruz, orientador da Dissertação, pela confiança, motivação, apoio e disponibilidade demonstrada, assim como pela notável capacidade de transmitir a sua sabedoria académica e humana que me guiaram até à concretização deste mestrado.

A todos os professores da comissão de curso deste mestrado, que se disponibilizaram com os seus conhecimentos e incentivos necessários ao desenvolvimento deste projeto.

Ao meu grande amigo Ricardo Balinha e pela cooperação prestada ao longo de todas as fases de desenvolvimento, nomeadamente na cooperação prestada na obtenção da informação necessária ao desenvolvimento deste trabalho revelando-se decisiva para o resultado final obtido.

A todos muito obrigados.



# 1. INTRODUÇÃO

A evolução da tecnologia de informação continua a determinar profundamente o modo como as organizações evoluem e os negócios que se desenvolvem. Um elemento intrínseco a qualquer organização é o seu sistema de informação, constituído por pessoas, procedimentos, aplicações informáticas, dado e equipamentos. O desenvolvimento tecnológico veio permitir que toda a informação possa ser suportada em computadores. Assim, ao nível das organizações, o sistema de informação tende a ter um suporte informático cada vez mais significativo.

A informação exige que sejamos capazes de descrever com rigor o modo como as nossas organizações funcionam, para que os sistemas de informação possam satisfazer plenamente as nossas necessidades. Este requisito é igualmente importante quer se venha a optar pela aquisição de uma aplicação informática existente no mercado ou por um desenvolvimento específico.

As aplicações informáticas dos nossos dias tendem a ser cada vez mais flexíveis, mas não estão preparadas para satisfazer todas as possibilidades de necessidades de informação dos seus potenciais utilizadores. Frequentemente somos obrigados à evolução da tecnologia de informação e esta continua a determinar profundamente o modo como as organizações evoluem e são decisivas no modo como os negócios se desenvolvem.

O desenvolvimento de sistemas informáticos depara-se com desafios semelhantes aos que se encontram noutras áreas de criação técnica, como a arquitetura ou a engenharia. Estes desafios talvez sejam maiores, já que os sistemas informáticos tornam-se mais complexos devido à sua permanente atualização, de forma a responder

às contínuas solicitações de melhoria colocadas pelos seus utilizadores e a tirar partido de novos desenvolvimentos tecnológicos.

Os Sistemas de Informação são produtos de software cada vez mais presentes e importantes no dia-a-dia das pessoas. Hoje em dia os SI existem num ambiente dinâmico e em mudanças contínuas, tendo um papel preponderante nas organizações, sendo determinantes para a sua capacidade competitiva.

As pessoas, os dados, as atividades ou recursos materiais têm cada vez mais a necessidade de um rigoroso alinhamento, no sentido de melhor gerir a informação, adotando as tecnologias de informação para o suporte às mudanças dos recursos envolvidos no planeamento, desenvolvimento e exploração de sistemas de informação.

Desenvolver software é uma atividade complexa por natureza. Uma das razões para esta afirmação é que não existe uma solução única para cada cenário de desenvolvimento. Além disso, lidamos todo o tempo com pessoas, o que torna o sucesso do projeto diretamente relacionado à dependência da competência da equipa de trabalho e às tecnologias de desenvolvimento utilizadas. E, para dificultar ainda mais, muitas vezes não fazemos uso de um processo bem definido para apoiar as atividades dos projetos.

## 1.1 A GESTÃO DE PROJETOS DE SOFTWARE

A gestão de projetos de software diz respeito ao alinhamento das necessidades dos clientes, com a boa gestão dos colaboradores participantes nas equipas de trabalho e a melhor administração do tempo e custo dos recursos do projeto.

É função da gestão de projetos de software garantir que o projeto cumpre os prazos a que se propôs cobrindo todos os requisitos para o qual foi definido, portanto, de acordo com as necessidades do cliente. Dada a sua complexa natureza de gestão, pois o software não é uma ciência exata, estando sujeito a constantes mudanças em termos de recursos, é obrigatória a flexibilidade de processos de desenvolvimento orientada às funcionalidades do projeto a desenvolver.

A gestão de projetos compreende métodos e ferramentas que organizam as tarefas, identificam a sua sequência de execução e dependências existentes, ajuda na atribuição de recursos além de definir patamares de restrições de execução servindo de guia para um grupo de trabalho.

Um plano de projeto é fundamental para evitar colapsos ao longo de todo o projeto, seja a nível da definição das atividades, seja ao nível dos “milestones” necessários à execução de todas etapas do projeto em execução.

Um bom desenvolvimento implica um bom planeamento. Este trabalho ainda continua a ser muitas vezes dispensado e por isso é que muitos dos sistemas de informação falham no cumprimento dos requisitos para os quais são projetados. Além disso o dilema do excesso de informação entre os vários intervenientes produtores de sistemas de informação e a falta de seleção qualitativa da mesma contribuem drasticamente para uma rutura nos projetos de software.

O dispensar de um bom planeamento por parte dos intervenientes de um projeto de software prende-se por vários fatores. A ainda falta de conhecimentos de algumas tecnologias e metodologias de suporte ao desenvolvimento, a ausência de ferramentas apelativas integradas que permitam fazer a monitorização planeada do sistema, através da criação de diferentes tipos de diagramas de modelação. A restrição e

categorização da informação para os diferentes intervenientes do sistema. A colaboração e valorização da informação inicial desprezada, mas muitas vezes influente para o sistema final. A definição correta da sequência das tarefas e os diferentes comportamentos do sistema que as mesmas envolvem, são fundamentais para o sucesso do desenvolvimento de software.

Há muitas empresas de desenvolvimento de software que evitam investir fortemente na modelação e planeamento de software integrado, sobretudo no que respeita à criação de diagramas de casos de uso, de atividades, de colaboração e sequência, interfaces etc. E muitas vezes ocorrem dependências incongruentes devido à sensibilidade ou insensibilidade por parte de um analista ou por parte de um programador. Embora haja linguagens de estruturação e modelação do negócio e do sistema, ainda escasseiam na interação com os intervenientes o que pode levar a um deficiente planeamento.

A gestão de projetos de software tem ganho cada vez mais adeptos, que sustentam o suporte básico para a eficiência e sucesso de um sistema de informação. Quando pensamos em software, pensamos em desenvolvimento de software, o que na realidade embora estejam intimamente ligados têm abordagens diferentes que convergem para a mesma finalidade.

Neste sentido é necessário o uso de métodos, técnicas e ferramentas de construção, para que possamos obter bons níveis de qualidade a custos controlados.

## 1.2 A TECNOLOGIA OPEN SOURCE

Para quem contacte com as Tecnologias de Informação, seja como utilizador, seja como profissional, é difícil nos dias de hoje ignorar o que se designa por “Open Source Software” (OSS).

“Open Source” é um termo inglês que significa código aberto, o qual pode ser partilhado ou até mesmo modificado devido à sua distribuição ser de acesso público. O termo foi criado pela OSI (Open Source Initiative) e desenvolvido por Eric Raymond e outros fundadores da OSI com a finalidade de apresentar o software livre a empresas evitando formalismos. Segundo a OSI, é preciso abranger dez pontos importantes para que um software possa ser considerado “Open Source”. Neste sentido, o software deve ser de distribuição livre, permitir o acesso ao código fonte bem como permitir a sua adaptação ou modificação, manter a identificação do autor do código fonte, não permitir a discriminação contra pessoas ou grupos, ou mesmo áreas de atuação, facultar a distribuição da licença não sendo esta específica a algum produto, ou que restrinja o uso de outros produtos e com licenças neutras em relação à tecnologia [Discover an open source, O que é open source].

O interesse nestas tecnologias tem-se massificado devido ao seu crescente alinhamento com as tecnologias fechadas, à sua franca integração operacional em todas as tecnologias aplicacionais e ao baixo custo que comporta a longo prazo. A título de exemplo, num estudo [Caso de estudo de sucesso da implementação Open Source, 2011] numa escola no norte do país, foi possível diminuir custos inerentes ao software com recurso à implementação de tecnologias Open Source.

Em Portugal, segundo estudos recentes, a adoção de estas tecnologias tem sido devidamente ponderada essencialmente por questões económicas. Num estudo levado

a cabo pela universidade de Coimbra, refere-se que Portugal poderia ter poupado cerca 113 milhões de euros entre 2008 e 2013 em software [Estudo de poupança com Open Source, 2014]. Num estudo baseado numa dissertação de um aluno da faculdade de economia do Porto que analisa a adoção do Open Source nos municípios portugueses [Rocha Paulo, 2012 – Análise da adoção do Open Source nos municípios portugueses], conclui-se que cerca de 45% dos municípios adotou o Open Source em ferramentas de tecnologia pelas mais variadas razões. O mesmo estudo refere que tem aumentado os adeptos destas tecnologias, sejam em Portugal, seja na Europa e fora dela.

Persistem, no entanto, alguns preconceitos relativos ao tema “Open Source Software” [Open Source Software – Que oportunidades em Portugal, 2004]. A ideia de que não haverá suporte para as tecnologias OSS, o custo real das soluções OSS e escassez de recursos humanos com formação especializada neste domínio, estão entre os principais entraves a esta tecnologia e por consequência lideram as estatísticas da deficiência de utilização.

Se por um lado deve ser ponderado e devidamente alinhado todo o processo de transição para esta tecnologia dentro de uma organização, começando pelas áreas menos críticas, a verdade é que depois de estabilizada, a médio prazo é possível transformar o investimento numa poupança efetiva, permitindo concorrer diretamente com as tecnologias que não são de mercado livre.

Neste contexto complexo em que estamos inseridos, o estudo e a compreensão da dinâmica de funcionamento de todo um projeto de software são fundamentais para suportar a concepção, a implementação e a operacionalização da maior parte dos processos de negócio do projeto que estamos a desenvolver.



Inspirado nesta dialética entre os sistemas “Open Source” e os sistemas de licença fechada relativos à gestão de projetos de software, proponho-me modelar e desenvolver um projeto de suporte à gestão de projetos de software para a web baseado em tecnologias “Open Source”.

### 1.3 APRESENTAÇÃO DE OBJETIVOS DO PROJETO

A importância de reduzir custos associados à produção de software, procurando também a redução de erros de integração entre os modelos de software e o planeamento e monitorização de tarefas para o seu desenvolvimento, exige uma boa definição de requisitos com diferentes graus de prioridade pelos diferentes intervenientes no projeto de software, a integração de modelos de sistemas alinhados com a sequência de tarefas que eles comportam e o acompanhamento integrado de todos os intervenientes no projeto. Isto é um problema latente na gestão de projetos de software.

Como referido, a gestão de projetos de software complementa várias etapas e é um processo cada vez mais complexo. Esta complexidade deve-se ao conjunto alargado de aspetos a considerar tais como planificação de cada tarefa a ser executada e seu ciclo de vida, o estudo evolutivo de projetos semelhantes na identificação de processos semelhantes e proposição de melhorias, bem como a real customização e planificação dos respetivos projetos em laboração. Esta complexidade aumenta quando o projeto necessita de estar disponível para vários intervenientes em múltiplos contextos, assumindo cada um deles um papel que pode tornar-se decisivo no resultado final.

A noção de contexto aumenta e caracteriza diferentes perfis de utilizador (com diferentes preferências ou mesmo diferentes intervenções na definição e modelação de requisitos) e diferentes plataformas e dispositivos.

Paralelamente a este problema, verifica-se que nem sempre há soluções a custo reduzido de ferramentas que permitam esta gestão e vão ao encontro das reais necessidades do cliente final.

Para minimizar este problema e detetar potenciais erros ou até evitá-los em casos futuros, seja no processo de análise ou na fase de desenvolvimento, os intervenientes devem participar ativamente contribuindo sempre que necessário com a sua opinião de forma a alinhar as necessidades do cliente com o desenvolvimento do projeto a ser concebido.

O objetivo é que o cliente final possa acompanhar todo o processo contribuindo com as suas perspetivas sobre a melhor forma de o software dar resposta às necessidades dos utilizadores. É neste contexto que a gestão do projeto tem necessidade de uma ferramenta de calendarização e acompanhamento de cada requisito no seu ciclo de vida ao longo de toda a vida do projeto.

Neste sentido, foram objetivos deste estudo: verificar que ferramentas existem para este suporte, quais as suas funcionalidades, que ambiente de interação apresentam aos utilizadores e que tecnologias dão suporte a essas ferramentas. Modelar e desenvolver uma aplicação web alternativa totalmente “Open Source”, que dê suporte a todas as etapas de definição, modelação e gestão de requisitos num projeto de software.

A ferramenta desenvolvida utiliza postgresQL 9.3 para o armazenamento de dados que torna-se rapidamente informação, Spring Tool Suite como editor e

ferramenta de desenvolvimento de toda a solução, sendo suportada por vários plugins e templates que usam a tecnologia de ponta. Neste caso concreto integra o Thymeleaf como framework HTML para os componentes da interface com o utilizador, jquery como framework no javascript para apresentar uma interface melhorada e menos subcarregada e o Foundation Template que de forma reconhecida aumenta a usabilidade interativa da interface.

A escolha destas ferramentas deveu-se ao facto de tratar-se de tecnologia “Open source”, utilizada no mundo empresarial e por isso muito abrangente, que permite desenvolver diferentes aspetos dos quais se pode salientar o bom funcionamento de ferramentas ORM e a integração de conceitos como o de Model-View-Controller (MVC) e Inversion of Control (IoC).

Embora nesta versão do projeto desenvolvido apenas estejam em funcionamento algumas das suas funcionalidades desejadas, o facto desta comportar vários cenários inovadores focada em diferentes aspetos, permite que este projeto possa evoluir para acomodar outras funcionalidades (por exemplo a monitorização do histórico de cada detalhe de cada projeto ou a impressão de relatórios em formato pdf).

## 1.4 ESTRUTURA DA DISSERTAÇÃO

Além do presente capítulo, esta dissertação encontra-se organizada em outros seis capítulos. No próximo capítulo aborda-se as tecnologias presentes na ferramenta desenvolvida e conceitos que elas comportam. No capítulo 3 são apresentados as principais ferramentas existentes que dão suporte à gestão de projetos de software bem como as suas características. O capítulo 4 consiste na apresentação da ferramenta

desenvolvida sendo exposto no capítulo 5 uma aplicação prática da abordagem definida no capítulo 4. Finalmente no capítulo 6, é efetuada a análise dos resultados do trabalho desenvolvido, apontando-se conclusões e propostas de trabalho futuro.

## 2. DEFINIÇÕES E CONCEITOS

### 2.1 INTRODUÇÃO

Com a popularização do Java em ambientes corporativos, a codificação das queries SQL e o respetivo código JDBC responsável por interagir com as mesmas era uma etapa que obrigava a um esforço extra, que resultava num problema de produtividade. Apesar de se tratar de uma linguagem de padrão ANSI, a codificação dependia do fabricante o que implicava um esforço alargado de implementações, apenas por se mudar do fabricante de base de dados.

Ao nível de paradigma também se fazia refletir o problema da mudança, este agora porque o paradigma da programação orientada a objetos difere muito do esquema de entidades relacionamento, pois a forma de pensar tinha de ser diferente para o mesmo sistema.

Para representarmos informações na base de dados, utilizamos tabelas e colunas. As tabelas geralmente possuem chave primária (PK) e podem ser relacionadas por meio da criação de chaves estrangeiras (FK) noutras tabelas.

Quando trabalhamos com uma aplicação Java, seguimos o paradigma orientado a objetos, onde representamos a informação através de classes. Nestas podemos ter atributos utilizando herança, conceito de polimorfismo, enumerações, entre outros.

Ou seja a todo momento deveríamos "transformar" objetos em registos e registos em objetos.

Durante o desenvolvimento de software, a evidente preocupação de se aumentar a produtividade aliada à qualidade e inovação tecnológica, quer no que se

refere às operações sobre a base de dados, quer à separação de componentes da arquitetura, recorreremos à utilização de Spring MVC.

O Spring MVC suporta a integração com Hibernate, Java Persistence API (JPA), e Java Data Object (JDO) para a gestão de recursos e implementações de Data Access Objects (DAO), bem como a possibilidade do uso do princípio Inversion of Control (IoC) e respectivo mapeamento de todos os recursos com a configuração de Object Relational Mapping (ORM) através de Dependencies injections.

## 2.2 A ARQUITETURA MVC

O MVC (Model-View-Controller) é uma arquitetura de sistemas de software para a construção de aplicações interativas complexas com interfaces gráficas, baseada em padrões (Patterns) como Observer, Composite e Strategy [Campozano Nelson, 2013].

O MVC, permite a separação de um projeto em múltiplas camadas, das quais fazem parte: Modelo(Model), Visão(View) e Controlador(Controller), correspondendo respetivamente aos padrões anteriormente designados (Ver Figura 1).

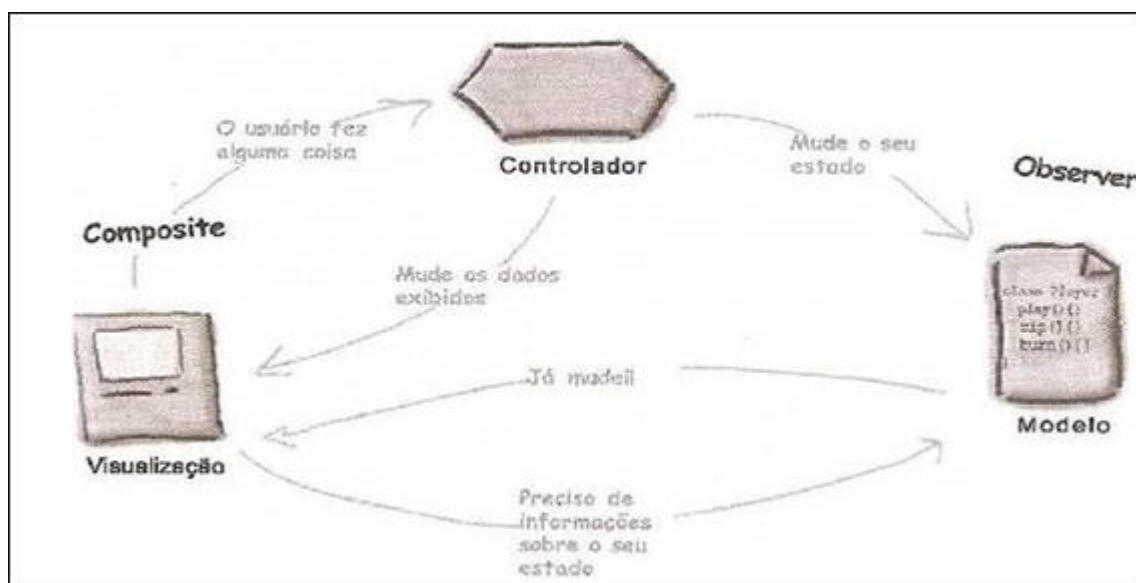


Figura 1 - Arquitetura MVC (FREEMAN & FREEMAN p. 424).

A camada Model (Ver Figura 2) é responsável pela lógica e regras de negócio. Encapsula o estado e o comportamento da aplicação e é a componente que serve de interface com a base de dados. Como as bases de dados são, na sua grande maioria, relacionais é necessário no modelo existir um mapeamento dos objetos do software orientado a objetos para as tabelas da base de dados. Além desta responsabilidade ela permite também o uso de padrão (DAO), Data Access Object [Padrões DAO], que permite melhor acesso aos dados por parte dos objetos. A camada Model é a combinação dos dados e dos métodos que os manipulam diretamente.

A camada View é responsável pela apresentação. É a interface de representação do modelo, ou seja, é a fronteira entre o utilizador e o sistema. Pode permitir ou ocultar a exibição de determinados atributos, atuando como filtro. As notificações de eventuais alterações são centralizadas no controlador, sendo esta apenas e só responsável pela apresentação dos resultados geridos pelo controller.

A camada Controller é responsável pelo controle e comunicação entre o modelo e a visão. Por ser considerado como uma fronteira entre os outros dois componentes do MVC a sua finalidade é controlar interações que ocorrem a partir da camada de visão (recebendo inputs) e “descobre” o que essas entradas significam para o modelo, os quais poderão ser eventos a despoletar métodos em resposta a determinados eventos.

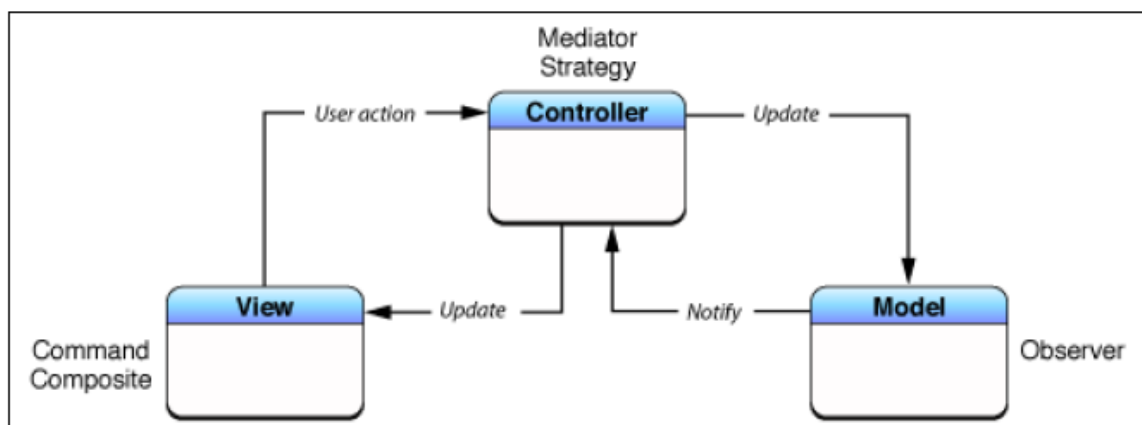


Figura 2 - Outro esquema da arquitetura MVC (retirado de [Arquitetura MVC]).

Em suma, o MVC separa a lógica de negócio no model, a apresentação na view e a interação entre eles no controller. A sua importância deve-se à agilização do desenvolvimento e respetiva manutenção, pois como é dividido em camadas Model, View, Controller, é mais fácil de fazer alterações. A manipulação do código também é beneficiada, podendo modificar-se qualquer camada minimizando a necessidade de modificar as outras.

## 2.3 O JAVA PERSISTENCE API E FRAMEWORK ORM

Para auxiliar a resolver o problema de mapeamento entre objeto-registo e registo-objeto, tornou-se popular em java e noutras linguagens, as ferramentas de mapeamento de objeto-relacional (ORM).

O Object-relational mapping é uma técnica de estruturação e desenvolvimento de software para reduzir o intervalo de abstração entre a programação orientada aos objetos e as bases de dados relacionais. Assim, as tabelas das bases de dados são representadas por classes e os registos das bases de dados são registados como instâncias das classes.

Com esta técnica o programador não tem que se preocupar com comandos em SQL, bastas apenas uma simples interface para fazer toda a persistência.

O Hibernate é uma ferramenta ORM Open Source e é a líder de mercado, sendo a inspiração para a especificação Java Persistence API (JPA). O Hibernate nasceu sem JPA mas hoje em dia é comum aceder ao Hibernate pela especificação JPA. Como toda a especificação, ela deve possuir implementações. Entre as implementações mais comuns, podemos citar: Hibernate do JBoss, EclipseLink da Eclipse Foundation e o OpenJPA da Apache.



O Hibernate abstrai o código SQL, toda a camada JDBC, e o SQL é gerado em tempo de execução. No caso das bases de dados, ele gera o SQL que serve para a respetiva base de dados, permitindo assim a troca de base de dados sem ter de se alterar código Java, já que isso é uma responsabilidade da ferramenta.

## 2.4 A FRAMEWORK SPRING MVC

A Framework Spring (Ver Figura 3) é uma framework de código aberto e inversão de controlo (IoC) e injeção de dependências. As funcionalidades da framework podem ser utilizadas sobre o Java EE, embora com extensões para os diferentes tipos de aplicações.

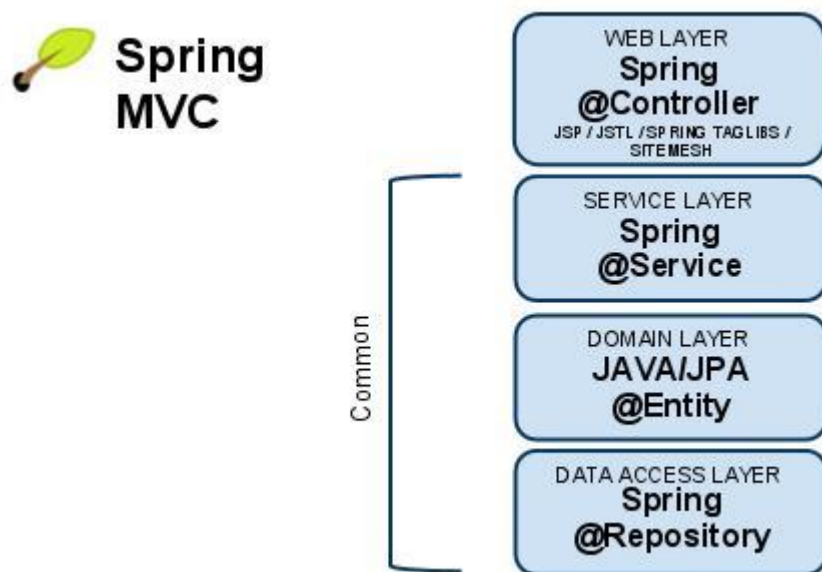


Figura 3 - Arquitetura Spring MVC (Spring Scaffolding Support in MyEclipse).

No Spring o container encarrega-se de "instanciar" classes da aplicação Java e definir as dependências entre elas através de um arquivo de configuração em formato XML, utilizando anotações em classes ou auto-wiring, métodos e propriedades. Neste sentido o acoplamento ou a dependência entre entidades ou classes da aplicação em

desenvolvimento é baixo, o que faz dele, uma boa prática de desenvolvimento ao nível estrutural e conceptual.

## 2.5 TECNOLOGIAS E CONCEITOS UTILIZADOS

Neste projeto foram utilizadas as seguintes tecnologias: Postgresql9.3 para a gestão de base de dados, StarUml e balsamiq Mockups na modulação de todo o sistema apresentado e a Spring tool Suite3.6 com Thymeleaf no desenvolvimento da aplicação.

### 2.5.1 *POSTGRESQL 9.3*

Postgresql 9.3 é um sistema de base de dados relacionais e objetos. É uma ferramenta “Open Source” disponível mediante uma licença de livre utilização. Este sistema é concorrente com outros sistemas de bases de dados livres tais como MySQL e FireBird bem como os proprietários como Oracle ou Microsoft SQL Server.

Este sistema de Gestão de base de dados relacionais e objeto, utiliza tipos de dados modernos, ou seja, suporta mais tipos de dados para além dos dados simples como os inteiros. Neste sentido o utilizador pode criar tipos de dados de funções ou até mesmo herdar de outros tipos de dados. É um sistema que funciona em diversos sistemas operativos. A título de exemplo, funciona sobre Solaris, Mac, Linux e Microsoft Windows para além de outros sistemas operativos.

Por ser estável, próxima da Oracle e por permitir possibilidades de programação dentro do motor da base de dados via PL/pgSQL escolhemos esta ferramenta.

### 2.5.2 *STARUML*

StarUml é um programa de modelação UML “Open Source” que permite gerar a maior parte dos diagramas específicos à modulação de um sistema de informação, bem como a sua exportação para vários formatos.

Através desta ferramenta é possível e de forma gratuita, definir o modelo de classes popularmente conhecido de modelo Entidades Relacionamentos, perceber, identificar e definir os casos de uso envolvidos nas diferentes interações no ceio do sistema em modulação.

A implementação e o desenvolvimento são importantes e aspetos a ter em atenção quando se define e modela um sistema cooperativo e complexo. O StarUml, permite antecipar a sua definição e redefinição, bem como a sua interação com os diferentes utilizadores do sistema com o recurso à definição de diagramas de colaboração, sequência e implementação.

### 2.5.3 *BALSAMIQ MOCKUPS*

Balsamiq Mockups é uma interface gráfica que permite antecipadamente, construir as interfaces da aplicação em desenvolvimento. Permite desenhar e arrumar todos os componentes “drag-and-drop” dando um aspeto próximo da realidade final.

A sua disposição gráfica é de fácil utilização e interativa permitindo ao utilizador inexperiente facilmente se adaptar e adota-la para apoiar o seu trabalho de desenvolvimento (Ver Figura 4).

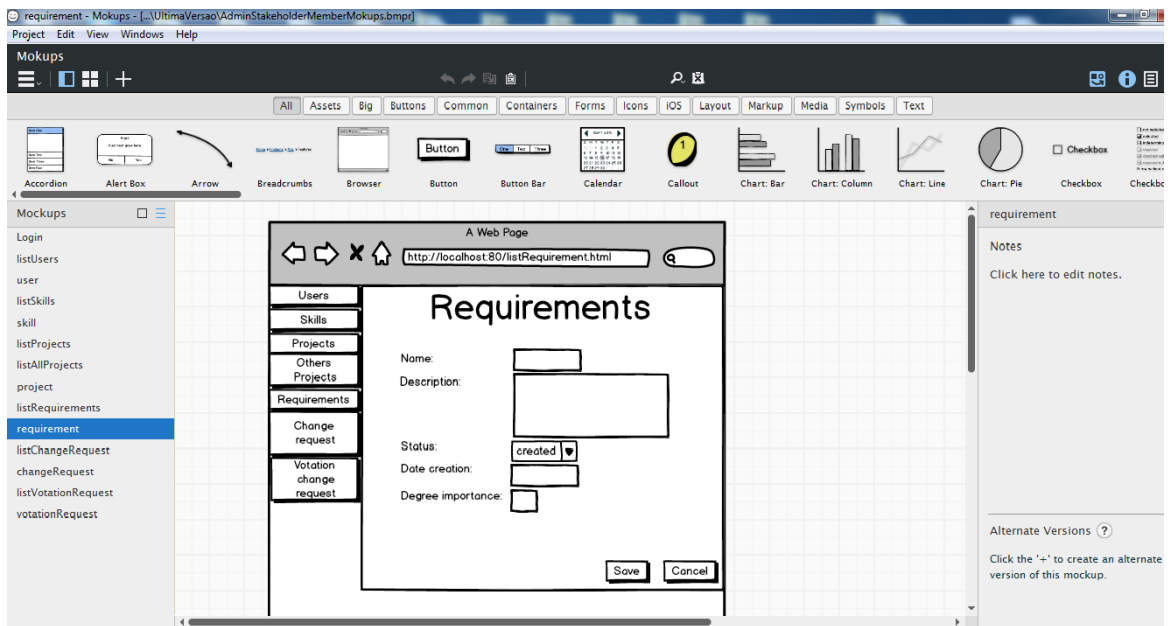


Figura 4 – Ambiente gráfico do Balsamiq mockups.

Esta não é uma ferramenta “Open Source”, mas permite a quando da sua instalação um período mensal experimental que convida a sua utilização sem recorrer a outro período alargado de utilização. Além disso, esta aplicação gráfica permite facilmente exportar todos os modelos desenhados para formatos reconhecidos como pdf, png ou xml. É possível exportar e transformar as mockups definidas diretamente em código html/css/js.

Neste projeto apenas recorri a esta ferramenta para definir as interfaces gráficas do sistema a desenvolver.

## 2.5.4 SPRING TOOL SUITE

O **Spring Tool Suite** (STS) disponibiliza um ambiente de desenvolvimento, baseado no IDE Eclipse, com um kit de ferramentas pré-instaladas para o desenvolvimento de aplicações em Java utilizando tecnologias do Spring, construída sobre eclipse.

Muitas tecnologias Spring são suportadas pela ferramenta, entre elas Spring Core, Spring Integration, Spring Batch, Spring WebFlow e Spring Data. Além do IDE, o kit inclui as versões mais atuais do Spring Roo, do Tomcat Server e do plugin para integração com o Maven.

Esta ferramenta inovadora está organizada por módulos o que permite para além do aumento da organização, o uso das funcionalidades do STS em outras ferramentas de forma individual como plugins.

O vasto conjunto de funcionalidades potencia de imediato a combinação e o suporte de diferentes frameworks de programação combinando-as com a linguagem Java, Web e framework JavaEE do eclipse.

Desde a sua instalação, o STS permite a sua gestão através da criação de diferentes tipos de projetos, dos quais se destaca a criação de projetos Maven Web.

O Maven é uma ferramenta de automação de compilação utilizada em projetos java, facilitando a sua gestão. Neste sentido, utiliza um ficheiro XML(POM) para descrever o projeto em desenvolvimento, as dependências sobre os módulos e componentes externos, gerando a ordem de compilação, diretórios e plugins necessários. Ele tem como principal objetivo a compilação de código e respetivo empacotamento e importação das bibliotecas necessárias de um ou mais repositórios. A sua utilização através do ficheiro POM, permite importar as releases (verões/atualizações) necessárias para o funcionamento da aplicação através da definição da dependência que a aplicação necessita.

Neste projeto foram definidas as dependências do Spring, Postgresql e Thymeleaf, respetivamente os plugins necessários ao funcionamento das diferentes camadas aplicacionais.

#### *2.5.5 THYMELEAF*

O Thymeleaf é um motor de templates, sob licença Apache, que gera XML, XHTML ou HTML5. O seu objetivo principal é gerar vistas em ambiente web baseadas no modelo MVC. É composto por diferentes dialetos, com sintaxe facilitada centrada nos atributos. A funcionalidade desta framework permite também ao utilizador criar os seus próprios dialetos e organizar as suas vistas através do conceito de fragmentos o que estende ainda mais a sua potencialidade.

#### *2.5.6 FOUNDATION TEMPLATE*

O Foundation Template é uma framework CSS construída com Sass (Syntactically Awesome Stylesheets), um pré-processador CSS poderoso, que nos permite desenvolver muito rapidamente o design funcional, construído em cima dos estilos iniciais. Com esta ferramenta podemos organizar e escrever o código CSS que podemos manter ao longo da aplicação. Existem também plugins JavaScript que fazem interações úteis ao desenvolvimento, favorecendo o uso de JQuery e que permitem a gestão e manuseamento de botões, barras de navegação, formulários, etc.

### 2.5.7 JQUERY

O JQuery é uma framework javascript desenvolvida para simplificar os scripts do lado do cliente que interagem com o Html. Esta framework permite criar animações, manipular eventos e desenvolver aplicações ajax dando um contexto dinâmico à aplicação sem ter de recarregar todos os dados a cada necessidade de atualização de um atributo. Favorecendo a reutilização de código e a redução do mesmo para a criação de tarefas semelhantes, o jquery tornou-se uma mais valia nas aplicações de hoje.

### 2.6 NOTAS FINAIS

Numa Era da informação é vital compreender as melhores práticas e aplicações na área dos sistemas informáticos para alcançar os seus objetivos pelo uso eficiente dos recursos disponíveis no mercado “Open Source”.

A utilização de tecnologias de última geração e a melhor prática em software possibilita a construção de aplicações complexas em prazos aceitáveis, capazes de responder a diferentes necessidades de informação.

O sucesso é garantido pela velocidade com que os conceitos são assimilados e pela rapidez com que se adaptam de acordo com as tomadas de decisão. Neste contexto a integração e a extensibilidade de cada tecnologia deve ser analisada como um benefício preponderante na interação com o ambiente do projeto a desenvolver, antes de ser alocada a tarefas de desenvolvimento.

A necessidade acrescida de ter um sistema flexível, aberto ao escalonamento, e extensibilidade que suporte funções de planeamento fornecendo informação útil em

tempo real para tomadas de decisão, dificultam a avaliação quantitativa e qualitativa dos benefícios oferecidos por cada tecnologia de informação adotada.

Tendo em conta os objetivos anteriormente definidos, foram escolhidas as tecnologias enumeradas nos subtemas anteriores por comportarem a redução de custos operacionais, a melhoria da produtividade ao nível do desenvolvimento, a estabilidade e segurança que suportam no mercado de trabalho, bem como o seu alinhamento com as tecnologias mais usadas no mundo empresarial.



### 3. ESTUDO DO ESTADO DA ARTE

#### 3.1 INTRODUÇÃO

As deficientes interpretações de informação por parte dos diferentes intervenientes no desenvolvimento de projetos de Software aliada à inconsistente comunicação entre os mesmos, aumenta a complexidade de gestão do projeto em desenvolvimento.

Sendo o desenvolvimento de software um problema complexo e suscetível a falhas, a melhor forma de comunicação é a utilização de uma linguagem comum entre todos os intervenientes, que permitem descrever corretamente o tipo de trabalho que se efetua [Rosenfeld Louis, Future Practice Interview].

É importante destacar que a grande maioria do insucesso dos projetos de software deve-se à deficiente comunicação entre os principais envolvidos no projeto (Stakeholders). Num projeto as questões principais que devem ser devidamente compreendidas e clarificadas são:

1. Que problemas precisamos de solucionar?
2. Que recursos necessitamos para resolver o problema?
3. Que recursos possuímos para poder solucionar o problema?
4. Quanto tempo dispomos para implementar a solução no projeto?

Para responder a estas questões é essencial poder contar com uma ferramenta multi-disciplinar, baseada nas boas práticas de gestão de projetos, e focada na integração do conhecimento sobre o projeto e no armazenamento e disponibilização de

informação aos *Stakeholders*. A ferramenta certa e os conhecimentos de boas práticas de gestão de projetos permitem envolver todos os intervenientes de forma colaborativa, e contar com a sua participação na descoberta de nova informação sobre o projeto.

É importante clarificar que a gestão de projetos de software não é um só projeto voltado para profissionais de engenharia. É necessário perceber a sensibilidade e a relevância que cada interveniente tem em cada desenvolvimento e que contributo poderá dar ao longo do projeto.

Dada a importância das atividades de gestão de projetos no desenvolvimento de um novo produto de software, é essencial que esta se possa basear em ferramentas Open Source, que integrem todas estas características, pois por vezes, a ausência de aplicações integradoras conduz a deficientes soluções.

Neste capítulo apresenta-se a revisão do estado da arte ao nível da gestão de projetos, sendo dada particular atenção a uma aplicação modelada para esta finalidade e concluída em grande parte como resposta às necessidades latentes visadas nestes sistemas.

### 3.2 ABORDAGENS EXISTENTES

Uma ferramenta de apoio à gestão de projetos deve permitir perceber todo o workflow de um projeto, possibilitando gerir com o melhor rendimento os seus diferentes recursos de acordo com as suas restrições.

Há ferramentas que apresentam soluções para parte deste problema, mas a dependência de um custo de licenciamento empurra as empresas para fora do núcleo de ponderação da utilização destas tecnologias.

A utilização deste tipo de software não exige conhecimentos práticos de programação propriamente dita, mas por outro lado obriga a um conhecimento profundo de todos os recursos a alocar bem como a exigente comunicação para com os diferentes intervenientes do desenvolvimento do projeto.

Partindo da análise da oferta que existe no mercado Open Source, no que respeita a ferramentas existentes para o suporte a todo o processo de gestão de projetos de software, existem várias aplicações que abordam diferentes aspetos, entre os quais o conceito de “Equipa de trabalho”, “Tarefas”, “Calendarização”, “Escalonamento”, “Perfis do utilizador” e “Recursos”, múltiplos ambientes entre outros. No remanescente desta secção iremos caracterizar algumas dessas aplicações.

**Open Project Foundation** – Uma ferramenta criada pela equipa do Open Project, para permitir a gestão de projetos de software no suporte à tomada de decisão. Depois de várias versões, a ideia essencial desta ferramenta é congregar num ambiente web a gestão de recursos durante todo o período de desenvolvimento do projeto. Permite a criação de projetos, definir datas marco para a execução de cada tarefa, bem como anexar ficheiros de apoio a cada tarefa. Nesta aplicação é possível ter uma visão geral do projeto na opção “Vista Global” bem como calendarizar todos os recursos, embora o conceito de recursos seja limitado. Por exemplo o conceito de categoria de tarefas não é distinguido, o conceito de “use case” ou de “Requisito” não são distinguidos bem como o conceito de perfis de *Stakeholders* dentro do mesmo projeto.

Em geral a sua funcionalidade (Figura 5) concentra-se nas tarefas a executar por cada participante, acompanhando-os do seu desenvolvimento. A partilha do conhecimento pelos utilizadores envolvidos é uma realidade embora não distinguem

diferentes graus de prioridade entre as tarefas, nem mesmo diferentes tipos de tarefas a executar. Não existe claramente o conceito de “Equipa de análise” nem o conceito de “Votação” sobre pedidos de alteração de requisitos ou outros parâmetros do projeto, permitindo que utilizadores com diferentes papéis possam dar o seu contributo de forma a poder criar condições de estabilidade e compreensão dos diferentes requisitos entre os membros da equipa.

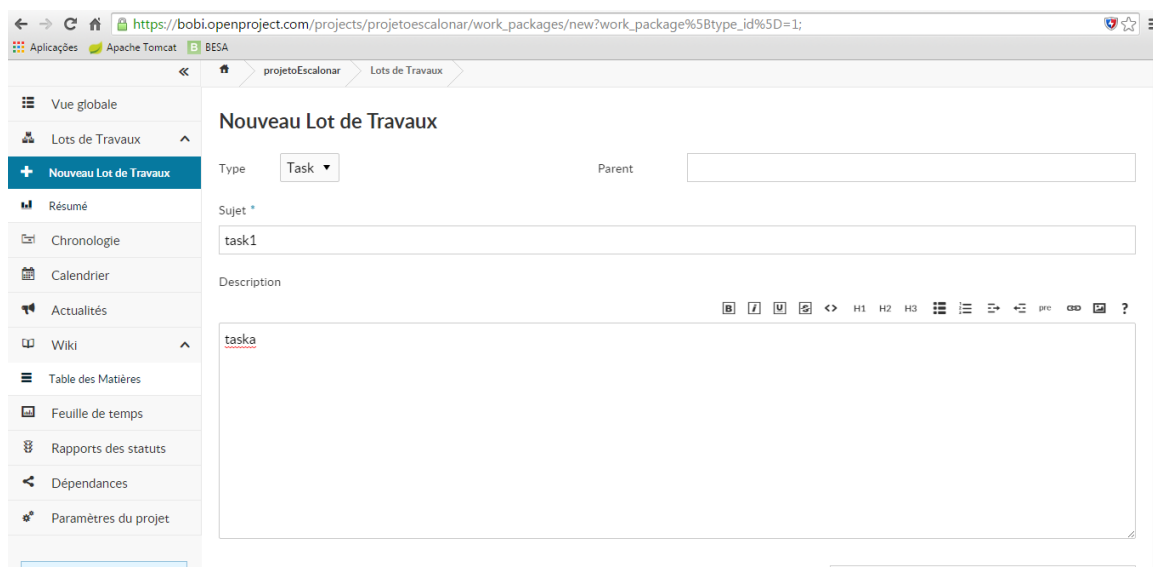


Figura 5 - Ferramenta de gestão de projetos da Open Project Foundation.

**Wrike** - Ferramenta em ambiente web suportada por vários sistemas operativos que permite alguma gestão de projetos, mas muitos dos conceitos, que a complexidade da gestão de projetos envolve, não são suportados. É possível criar informação relativa à gestão de um projeto em desenvolvimento e partilhar algumas discussões alargadas a diferentes utilizadores importando por isso o conceito de partilha de informação e favorecendo o alinhamento de uma linguagem comum para os diferentes participantes.

Não há propriamente o conceito de multi-projetos em gestão, mas tarefas escalonadas em calendário.

Ao nível da distinção de tarefas não há diferenciação entre as várias, pelo que o conceito de “Use Case” e “Requisitos” são integrados nas tarefas a executar expressamente no prazo definido.

O conceito de equipa de trabalho não existe na sua globalidade pelo que todos os utilizadores intervenientes são considerados elementos participantes na equipa de trabalho, o que na realidade não é uma boa prática devido ao risco acrescido de haver alguma incongruência entre os diferentes intervenientes.

Wrike é, portanto, uma ferramenta limitada, que não pode suportar todo o workflow que a gestão de projetos de software comporta sendo exigido um pagamento acima de 5 utilizadores por projeto, o que apresenta uma barreira financeira para potenciais clientes.

A sua disposição gráfica é simples possuindo uma excelente arrumação funcional relativamente às funcionalidades disponíveis a cada utilizador (Figura 6).

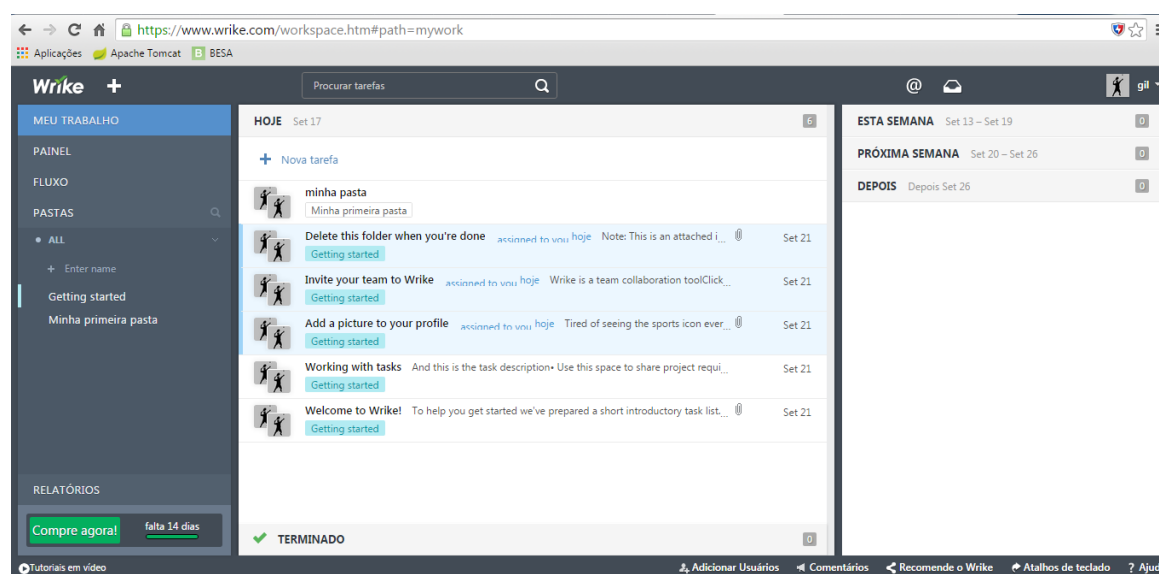


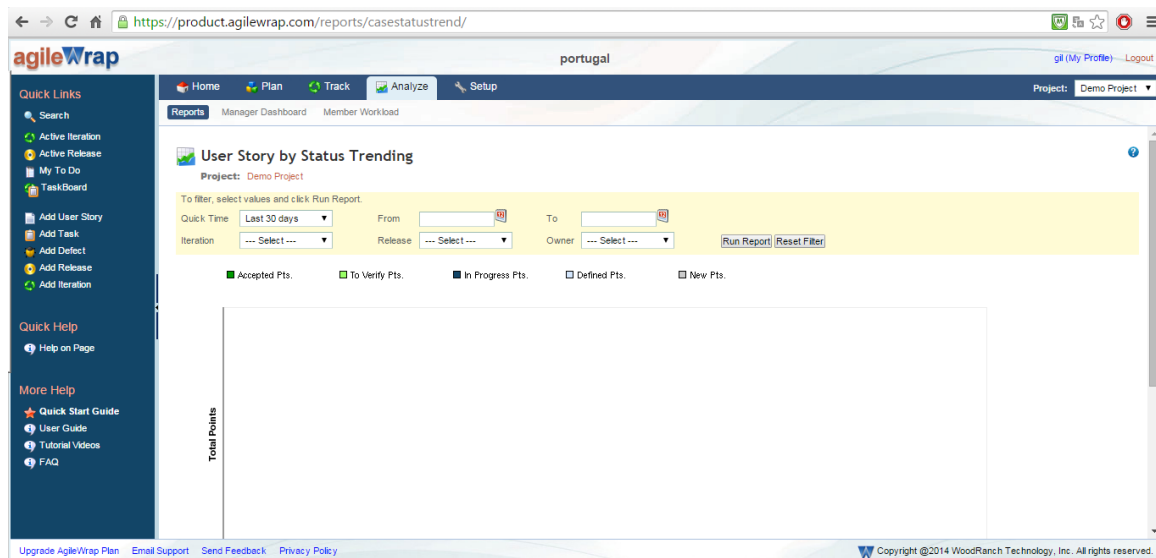
Figura 6 - Ferramenta Wrike para gestão de projetos.

**AgileWrap** – Ferramenta em ambiente web e disponível em vários sistemas operativos, gratuita até cinco utilizadores, que dá ênfase à gestão de projetos concentrando o conceito de “multi-projetos vs multi-colaboradores” (Figura 7). A sua função principal foca-se no detalhe das funções ou tarefas a serem realizadas pelos diferentes participantes. Neste sentido proporciona a gestão e alocação de recursos calendarizados pelas funções a desempenhar pelos diferentes *Stakeholders* (Colaboradores), embora não exista o conceito de “equipa de análise” nem a distinção de diferentes tipos de tarefas.

Desenhada para permitir reduzir custos, favorece o controlo das tarefas permitindo elaborar gráficos de apoio à tomada de decisão, calculando antecipadamente o risco e as tarefas que se atrasam no desenvolvimento.

Ao nível da gestão global, a ferramenta AgileWrap permite acoplar várias tarefas com gestão de prioridades por utilizador, permitindo seguir de perto tarefas a executar com maior urgência em cada projeto.

Em síntese, esta ferramenta favorece a gestão de projetos ao nível dos recursos alocados, não permitindo a participação de cada *Stakeholder* em tarefas que poderão provocar questões de incongruência e instabilidade. Falta, portanto, o conceito de “Votação”, o qual permitiria a cada membro de uma equipa de trabalho participar ativamente com o seu ponto de vista em determinadas tarefas.



**Figura 7 - Ferramenta AgileWrap para gestão de projetos.**

Para além destas aplicações Open Source de apoio à gestão de projetos, poderiam ser mencionadas outras, como o LibrePan e o ProjectOpen, também de código aberto. A maioria destas ferramentas são integradas noutras ferramentas ou integram funcionalidades acoplando outras aplicações no apoio ao seu desenvolvimento. Por exemplo, os gráficos de Gant são utilizados em todas estas aplicações, enriquecendo-as com uma funcionalidade intuitiva e cooperativa.

### 3.3 A IMPORTÂNCIA DO CONHECIMENTO NAS APLICAÇÕES DE GESTÃO DE PROJETOS

Cada vez mais a tecnologia é um benefício acrescido numa empresa, podendo este derivar de aplicações de metodologias ou técnicas de gestão de projetos, de forma a conduzir melhor, o seu planeamento.

A utilização correta das ferramentas certas pode criar oportunidades para as empresas focarem as questões essenciais ou consideradas mais críticas [CLARK, 1999].

A importância das tecnologias de informação é indiscutível, por isso surge a necessidade de realizar um planeamento adequado para os respetivos projetos. Metodologias eficientes de gestão de projetos de software são cada vez mais uma

necessidade e fazem parte integrante do conhecimento de uma empresa. Podemos considerar fundamentalmente a utilização de metodologias e técnicas já difundidas na gestão de projetos, pois estes já mostram níveis de eficácia incontornável.

Neste sentido a gestão de projetos nas tecnologias de informação baseia-se em conceitos elaborados no compromisso e com uma estrutura bem definida. Neste trabalho abordarei a aplicação de gestão de projetos segundo o modelo PMBOK (Project Management Body of Knowledge).

Uma série de modelos de processo de gestão de projetos de software tem sido elaborada nos últimos anos. Alguns desses modelos são originais e outros são derivados e/ou evoluídos de outros modelos. Para este projeto, foram estudadas as técnicas e ferramentas do guia de boas práticas Project Management Body of Knowledge (PMBOK), definidas pelo PMI. O PMBOK é um guia reconhecido de gestão de projetos que descreve normas, métodos, processos e práticas estabelecidas para gerir projetos de qualquer natureza [PMBOK Guide, 2013]. Os principais objetivos do PMBOK são as boas práticas da aplicação das ferramentas e técnicas para aumentar a probabilidade de sucesso de um projeto e o vocabulário comum utilizado pelos profissionais da área de gestão de projetos.

### 3.4 O PMBOK E A IMPORTÂNCIA DO CONHECIMENTO NA GESTÃO DE PROJETOS

O PMBOK é um documento que descreve técnicas, métodos e processos relativos à Gestão de Projetos. Neste sentido, gestão de projetos é “a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de alcançar os seus objetivos [Philips Joseph, 2002]”.



O PMBOK é reconhecido como um Padrão Nacional Americano pelo ANSI. A quinta edição é o padrão ANSI/PMI 99-001-2013 e teve alinhamento com a norma internacional ISO 21500:2012. Nele são formalizados diversos conceitos de gestão de projetos, como a própria definição de projeto e do seu ciclo de vida. O PMBOK é amplamente reconhecido como guia de boas práticas, aplicáveis à maioria dos projetos. Estes conhecimentos estão categorizados em dez áreas de conhecimento e os processos relacionados são organizados em cinco grupos ao longo do ciclo de vida do projeto. Mesmo não sendo uma metodologia de gestão de projetos, o PMBOK incorpora processos e atividades que suprem as necessidades de todas as etapas ou fases do ciclo de vida de um projeto. Dessa forma é plausível a adoção do PMBOK no planeamento dos recursos humanos, riscos, custos e *Stakeholders*. Outro fator que impulsionou a utilização deste corpo de conhecimento é sua característica genérica e aplicada a todos os tipos de projetos, inclusive projetos de desenvolvimento de software.

A importância do conhecimento envolve diferentes aspetos na gestão de projetos, interessando perceber e congregar a integração de portfolio, programas e projetos. Um portfólio refere-se a um conjunto de projetos, programas e subportfólios e operações de gestão que envolvem o grupo para conseguir atingir o objetivo que se propôs. Um programa envolve o agrupamento de projetos e relação com outros programas que o poderão apoiar na gestão do trabalho. Projetos são considerados parte integrante de um programa que podem ou não ter relação com os diferentes portfólios (Ver Figura 8).

Na organização de um plano estrutural é importante definir uma estratégia, a qual, a diferentes graus de prioridade, tem relação com os diferentes projetos individuais. Neste sentido o impacto da planificação do projeto é determinante para o

controle dos riscos e levar a bom porto o projeto a desenvolver. É nele que reside a gestão de recursos de suporte às componentes a definir, os riscos em cada componente e as várias linhas de negócio tais como a infraestrutura e processos de melhoramento ou eficiência.

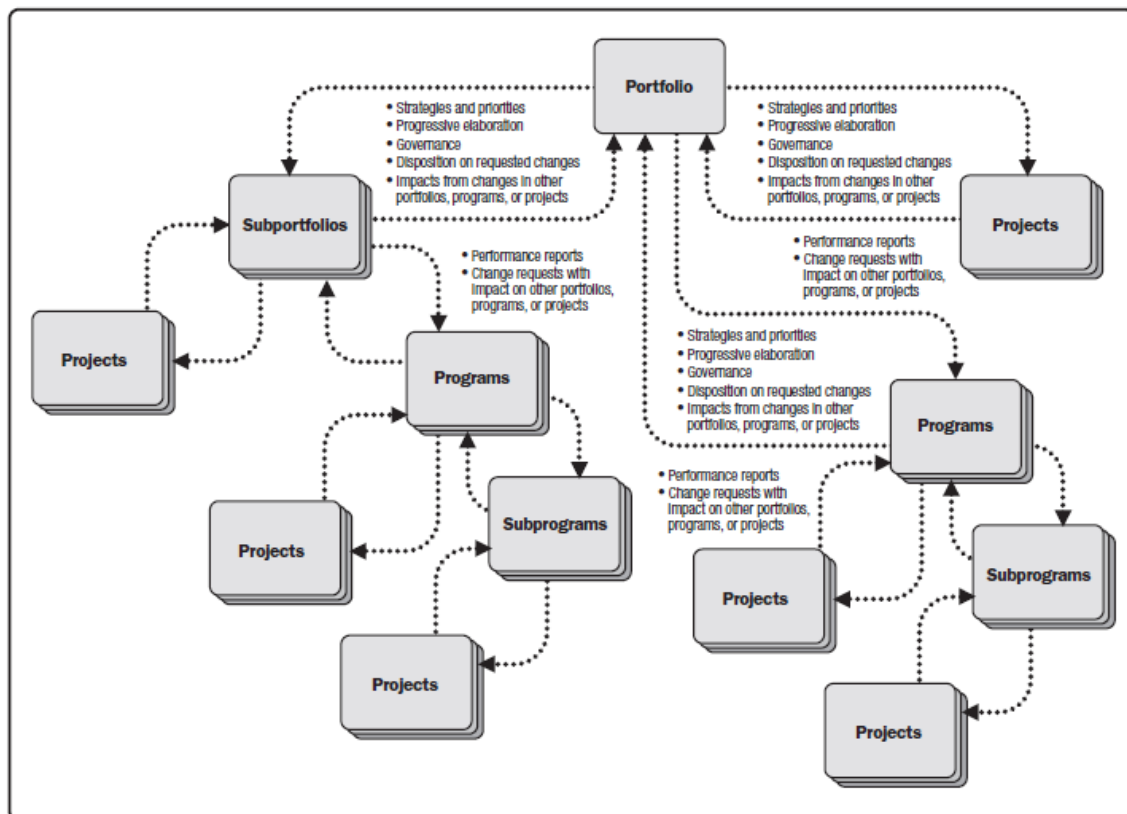


Figura 8 - Portfólios, Programas e Projetos (retirado de PMBOK guide, 2013).

A gestão de projetos é a gestão do conhecimento, habilidades, ferramentas e técnicas para que a equipa envolvida no projeto possa identificar os requisitos corretamente, planear tarefas, custo e prazos para a execução do projeto, e para que o gestor do projeto e outros *Stakeholders* possam monitorizar as tarefas planeadas para o projeto e aferir a saúde e progresso do projeto. Ao nível aplicacional são integradas um conjunto de regras baseadas neste modelo (PMBOK) que permitem a sua aplicação adequada.

### 3.4.1 CICLO DE VIDA DE UM PROJETO

O ciclo de vida do projeto define as fases que medeiam o início de um projeto e o seu final [PMBOK Guide, 2013]. O conjunto de fases de um projeto é conhecido como ciclo de vida do projeto. As fases do ciclo de vida do projeto dependem intimamente da natureza do projeto. Um projeto é desenvolvido a partir de uma ideia, progredindo para um plano, que por sua vez é executado e concluído. O projeto é dividido em fases que formam o ciclo de vida. Pode ser dividido e moldado de acordo com o projeto e a equipa de trabalho que o compõe, mas todo o projeto deve ter início e fim bem definidos.

Segundo o guia PMBOK, são considerados cinco grupos de processos na gestão de um projeto (Figura 9), sendo um processo o conjunto de ações e atividades interrelacionadas que são executadas para alcançar um objetivo. Cada processo é caracterizado pelas suas entradas, ferramentas e técnicas que podem ser aplicadas, e as saídas dos resultados [PMI, 2012]



Figura 9 - Grupos de Processos de Gestão de Projetos (retirado de Barbose Fernando, Gerenciamento de Projetos).

1. O grupo de processos de Iniciação trata da parte inicial do projeto, ou fase do projeto, onde é definida a missão e o objetivo do projeto ou fase.
2. O grupo de processos de planeamento inclui os processos destinados a identificar e selecionar as melhores entradas de abordagens do projeto, sendo planeado e detalhado tudo aquilo que será realizado, e qual o esforço e custo envolvidos.
3. O grupo de processos de execução materializa tudo aquilo que foi planeado anteriormente, sendo que qualquer erro ocorrido em etapas anteriores pode influenciar toda a execução do projeto, uma vez que a maior parte do orçamento e do esforço do projeto são consumidos nesta etapa. Nestes processos são mobilizadas e coordenadas pessoas e recursos necessários à execução do projeto.
4. O grupo de processos de monitorização e controlo arranca durante o planeamento operacional e prolonga-se durante toda a fase de execução do projeto. Tem como objetivo acompanhar e controlar aquilo que está a ser realizado pela equipa de projeto, de modo a corrigir e prevenir no mínimo espaço de tempo possível anomalias ou imparidades que poderão surgir.
5. O grupo de processos de finalização, ou fecho do projeto, ocorre quando a execução dos trabalhos é avaliada por auditoria interna ou externa, confirmando/validando que todo o desenvolvimento vai ao encontro das necessidades do projeto e do plano estabelecido para o mesmo.

### 3.4.2 ÁREAS DE CONHECIMENTO NA GESTÃO DE PROJETOS

O PMBOK entende a gestão de projetos como utilizando dez áreas base de conhecimento. São elas, gestão do âmbito, do tempo, do custo, da qualidade, dos recursos humanos, da comunicação, de riscos, das aquisições, da integração e de Stakeholders (Figura 10).

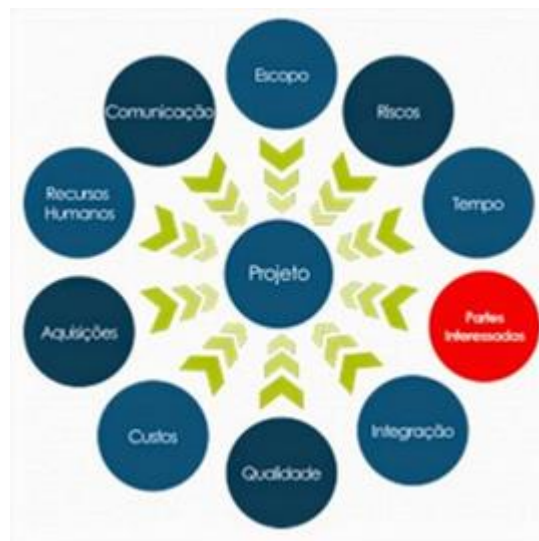


Figura 10 - Áreas de conhecimento do PMBOK (retirado de PMBOK guide, 2013).

O âmbito é a definição do resultado que se espera do projeto. O grande problema da gestão do âmbito para as tecnologias de informação é que o cliente é inconstante e muitas vezes não sabe bem o que precisa, outras vezes não sabe comunicar o que necessita ou pretende do projeto, sendo assim necessário que a equipa de trabalho saiba identificar essas necessidades da melhor maneira possível, construindo modelos a validar com o cliente à medida que forem definidos. No final do levantamento dos requisitos é importante formalizá-los em contrato a fim de evitar futuros problemas no final do projeto.

1. A gestão de projetos de software deve-se focalizar na gestão do âmbito, pois muitas vezes os clientes precisam de mudar requisitos durante a execução do

projeto e, caso o âmbito não esteja definido com clareza, outras tarefas referentes a requisitos já definidos podem acrescentar novos custos não previstos inicialmente nem orçamentados para o projeto.

2. A gestão do tempo ou prazo, corresponde ao estabelecimento de um cronograma do projeto, organizando cronologicamente as atividades e recursos. Este é um ponto crítico para a gestão de projetos, uma vez que os clientes procuram soluções com prazos curtos. Neste sentido, qualquer atraso pode resultar no fracasso do projeto e portanto, é importante separá-lo em etapas de entrega ao cliente final podendo validar o resultado gradualmente, minimizando o retrocesso.
3. A gestão de custos define o orçamento estimado e assume o controlo de custos.
4. A qualidade tem como objetivo, garantir a satisfação do cliente através de melhorias contínuas no seu processo. Neste sentido a gestão da qualidade assume um papel preponderante, podendo sofrer cortes de acordo com o prazo e custo do projeto. Num projeto de software, validações com os clientes, apresentando protótipos e diagramas, além das entregas faseadas do projeto são boas práticas e garantias extras do sucesso do projeto.
5. A gestão de recursos humanos do projeto, implica a formação de uma equipa responsável pela execução. Nesta fase é importante gerir de perto as expectativas a fim de manter o projeto controlado até à sua conclusão. Atualmente, é comum os recursos humanos de um projeto de software terem formação académica, permitindo ao gestor de projeto dispor de recursos especializados, dando mais confiança para a execução do projeto.
6. A comunicação é indispensável para qualquer projeto. Neste sentido é necessário identificar quais as informações que devem ser adquiridas ou levadas em conta

pelos diferentes membros, bem com saber quando e como. Devem ser definidos métodos de formalização para promover a comunicação, tais como reuniões, relatórios periódicos etc. Na gestão de projetos de software, a comunicação tem sido bastante otimizada com a utilização de videoconferências, emails, chats entre outros. No entanto é preciso ter muita atenção, porque um plano de comunicações mal elaborado pode comprometer a execução do projeto.

7. O plano de risco de um projeto deve ser levado em consideração. Todos os riscos ou potenciais riscos devem ser objeto de identificação, bem como as medidas para a sua prevenção ou contenção. Na gestão de projetos de software, muitas vezes o controlo de riscos não é devidamente valorizado devido aos prazos muito restritivos. Porém, pode ser um fator que conduz ao fracasso do projeto, pois a tecnologia está em constante evolução e muitas vezes, antes de se finalizar os projetos, uma nova tecnologia pode aparecer e revolucionar boa parte do trabalho realizado, apresentando-se mais eficiente no longo prazo.
8. A gestão de aquisições do projeto engloba o controlo de contratos e mudanças. Na gestão de projetos de software é indispensável um controlo rigoroso de contratos, garantindo um alinhamento de expectativas entre as partes interessadas, para evitar problemas no decorrer do projeto, assegurando a sua entrega conforme o planeado. Relativamente às aquisições, é necessário analisar quais os recursos que serão necessários, para planear a correta aquisição.
9. A gestão da integração consiste em manter coordenados todos os processos de forma coesa e sistémica. Na gestão de projetos de software a integração é essencial para que todas as áreas do projeto estejam em constante sintonia e

atualização, garantindo a execução do planejamento e realizando a entrega conforme o esperado.

10. A gestão de Stakeholder's caracteriza-se por identificar as partes interessadas, planejamento, envolvimento e controle das mesmas no projeto. Neste sentido esta área de conhecimento preocupa-se com o envolvimento e impacto que os interessados terão no projeto, sendo necessário definir estratégias para quebrar resistências garantido o envolvimento comum num todo. Nesta área é importante garantir a comunicação e privilegiar a comunicação de todos os intervenientes de forma a solucionar as questões quando ocorrem. Neste sentido é necessário monitorizar relacionamentos entre as partes, ajustar estratégias adequadas, eliminar resistências e aumentar o suporte ao projeto.

As características e circunstâncias podem influenciar as restrições específicas do projeto no qual o gestor de projeto precisa de definir a equipa de trabalho bem como as competências de cada colaborador na equipa ou fora dela. Convém nunca descurar que o projeto deve integrar todos estes fatores onde os diferentes colaboradores assumem um fator chave em toda a estrutura de gestão. Nomeadamente é importante valorizar que partes interessadas no projeto podem ter diferentes ideias a respeito de determinado requisito, atribuindo maior ou menor importância. Alterando por exemplo os requisitos ou objetivos do projeto podem criar-se riscos adicionais. Neste sentido a equipa de projeto precisa de ser capaz de avaliar a situação, equilibrar o pedido estabelecido e manter uma comunicação pró-ativa com as partes interessadas a fim de entregar um produto ou serviço do projeto com qualidade, dentro do orçamento e do prazo, podendo considerar o projeto bem sucedido.



### 3.4.3 Os 47 PROCESSOS DO PMBOK

Os grupos de processos da gestão de projetos têm grande correspondência com o conceito do Ciclo PDCA (**Plan - Do - Check - Act**) [PMBOK Guide, 2013]: Planejar - Fazer - Verificar - Agir. O grupo de processos de Planejamento corresponde ao Planejar; o grupo de processos de Execução, corresponde ao Fazer; a monitorização e Controlo engloba verificar e agir. E como a natureza dos projetos é limitada no tempo, o PMBOK ainda caracteriza os grupos de processos que iniciam (Iniciação) e finalizam (Encerramento) um projeto.

Para além dos conceitos e aspetos fundamentais da gestão de projetos, o guia PMBOK define e descreve processos de gestão de projetos organizados por área de conhecimento. Em cada processo, são abordadas as entradas e saídas, suas características, bem como as técnicas e ferramentas envolvidas.

São 47 os processos que o Guia propõe para a boa gestão de processos de um projeto, distribuídos pelos cinco grupos de processos anteriormente apresentados e as respetivas áreas de conhecimento associadas a cada um. Na Tabela 1 podemos ver todos os processos distribuídos por grupos de processos e áreas de conhecimento.

**Tabela 1 - Grupos de processos e Áreas de conhecimento do projeto (retirado da revista mundoPM, 2015).**

Área do conhecimento	Grupos de processos de gestão do projeto				
	Iniciação	Planeamento	Execução	Controle	Finalização
1. Gestão da integração do projeto	1.1 Definir termo de abertura do	1.2 Desenvolver plano de Gestão de projeto	1.3 Dirigir e orientar o trabalho do	1.4 Controlar o trabalho de projeto	1.6 Finalizar o projeto

	projeto		projeto		
				1.5 Realizar controle integrado de mudanças	
2. Gestão do escopo do projeto		2.2 Planejar a gestão de escopo		2.5 Validar escopo	
		2.3 Definir requisitos		2.6 Controlar escopo	
		2.4 Definir escopo			
		2.5 Criar EAP			
3. Gestão do tempo do projeto		3.1 Planeamento da gestão do tempo		3.7 Controlar cronogramas	
		3.2 Definir atividades			
		3.3 Sequenciar atividades			
		3.4 Estimar recursos das atividades			
		3.5 Estimar duração das atividades			
		3.6 Desenvolver cronogramas			
4. Gestão do custo do projeto		4.1 Planejar gestão de custos		4.4 Controlar custos	
		4.2 Estimar custos			
		4.3 Determinar orçamento			
5. Gestão da qualidade do projeto		5.1 Planejar gestão da qualidade	5.2 Realizar garantia da qualidade	5.3 Controlar qualidade	
6. Gestão dos recursos humanos do projeto		6.1 Planeamento da gestão de recursos humanos	6.2 Adequirir a equipa de projeto		
			6.3 Definir a equipa de projeto		
			6.4 Gerir a equipa de projeto		
7. Gestão da comunicação do projeto		7.1 Planejar a gestão da comunicação	7.2 Gerir a comunicação	7.3 Controlar a comunicação	

8. Gestão dos riscos do projeto		8.1 Planear a gestão de riscos		8.6 Controlar riscos	
		8.2 Identificar riscos			
		8.3 Realizar análise qualitativa dos riscos			
		8.4 Realizar análise quantitativa aos riscos			
		8.5 Planear respostas aos riscos			
9. Gestão das aquisições do projeto		9.1 Planear a gestão de aquisições do projeto	9.2 Conduzir aquisições	9.3 Controlar aquisições	9.4 Encerrar aquisições
10. Gestão dos Stakeholders do projeto	10.1 Identificar Stakeholders	10.2 Planear a gestão dos stakeholders	10.3 Gerir Stakeholders	10.4 Controlar Stakeholders	

#### 3.4.4 A ESTRUTURA DE GESTÃO DE PROJETOS (PMO)

O Project Management Office (PMO) é uma estrutura de gestão que padroniza a governança dos processos do projeto, facilitando a partilha de recursos, metodologias, ferramentas e técnicas. A responsabilidade de um PMO pode variar fornecendo funções de suporte à gestão do projeto. Existem vários tipos de estruturas de PMO, e cada uma varia com o grau de influência que tem em cada projeto, tais como:

- Suporte (PMOs) – fornece um papel de consulta para projetos, fornecendo modelos, formação, acesso a informação e aprendizagem com a realização de outros projetos.

- Controlador (PMOs) - Funções de controlo que exigem o cumprimento de determinadas estruturas recorrendo à metodologia específica utilizada. Formulários e ferramentas em conformidade com a governança.
- Diretiva (PMOs) – Funções assumidas pelo gestor de projeto a partir do controle efetuada pelos demais PMOs.

O PMO integra dados e informação de projetos e avalia ao nível superior quais os objetivos a atingir e quais estão a ser cumpridos. Neste sentido o PMO é a ligação natural entre o portfólio da empresa, os programas e os sistemas de medição do risco.

Um PMO pode ter autoridade para fazer parte integrante da equipa chave de decisão ao longo do desenvolvimento. Pode por isso recomendar ou sugerir requisitos conforme a autoridade que lhe for dada.

Num projeto de gestão, as operações a realizar são diversificadas e dependendo da estratégia definida, o chefe de projeto toma a sua organização.

É importante não descurar o valor do negócio de forma a poder acrescentar valor a cada projeto que se encontra em fase inicial, permitindo que o conhecimento adquirido esteja presente quer seja pelos recursos disponibilizados quer seja pelos objetivos atingidos.

#### *3.4.5 A CERTIFICAÇÃO PROJECT MANAGEMENT PROFESSIONAL(PMP)*

O PMP é um certificado emitido pelo PMI aos profissionais que satisfazem todas as condições do programa de certificação. A certificação PMP é vista por muitas empresas como uma evolução da carreira profissional e cada vez mais são exigidas para

o mercado de trabalho. A partir dessa certificação o profissional aperfeiçoa o conhecimento em gestão de projetos com base no guia PMBOK.

Para manter a certificação, é necessário o comprometimento do profissional que, segundo o PMIMG (2011), deve satisfazer as condições do Programa Contínuo de Requisitos de Certificação (Continuing Certification Requirements Program – CCR) do PMI. Esse programa baseia-se na acumulação de unidades de desenvolvimento profissional (Professional Development Units – PDU), sendo que um PDU tipicamente equivale a uma hora de formação planeada e estruturada ou de atuação profissional na gestão de projetos.

#### *3.4.6 GESTOR DE PROJETO E AS PARTES INTERESSADAS DO PROJETO*

A gestão de projetos implica a envolvimento de diferentes tipos de recursos humanos com diferentes funções. Dependendo das funções, o colaborador participa ativamente no projeto sem ter de se preocupar com aspetos que não lhe dizem respeito. Dentro deste quadro, a gestão de projetos comporta diferentes papéis (*Roles*): Gestor do Projeto (Project Manager), membro de equipa (*Team Member*), e outros intervenientes, onde se incluem *Other Stakeholders*.

Project Manager – é a pessoa designada pela empresa para liderar a equipa que é responsável para a execução dos objetivos do projeto. O papel do gestor de projeto, envolve por isso um conjunto de atividades que por si só, são de grande complexidade e de muita responsabilidade.

Em geral o Project Manager tem a responsabilidade de satisfazer as necessidades das tarefas, as necessidades da equipa e as necessidades individuais. Um Project

Manager tem de ter disciplina, estratégia crítica, boas capacidades de comunicação e compreensão entre os diferentes membros da equipa de forma a promover o desenvolvimento. Além de todas estas características é indispensável que o Project Manager tenha conhecimento e experiência na área que está a gerir e performance ao nível da gestão de recursos humanos. Neste sentido o gestor de projeto tem de possuir um conjunto de habilidades (*Skills*) onde se destacam a liderança, desenvolvimento de equipa, motivação, comunicação, influência, capacidade de decisão, confiança, coaching, gestão de conflito etc.

De todas estas habilidades é importante para um gestor de projeto saber gerir o planeamento dos interessados (*Stakeholder*), porque as suas características determinam a estratégia a adotar para aumentar o apoio, reduzir as resistências e minimizar os impactos negativos com origem nos grupos de interessados durante todo o ciclo de vida do projeto.

A definição da estratégia de gestão dos interessados obriga a:

- Identificação dos interessados que podem afetar o projeto de modo significativo.
- Identificação dos impactos (positivos e negativos) que o projeto tem em cada interessado ou grupo de interessados.
- Identificação das motivações e das expectativas que cada interessado ou grupo de interessados tem em relação aos resultados do projeto.
- Identificação do nível de participação pretendido para cada um dos interessados ou grupos de interessados.

As partes interessadas, designadas por *Stakeholders*, são pessoas, grupos ou organizações que podem afetar, serem afetados ou sentirem-se afetados por uma decisão, atividade ou resultado de um projeto. Elas englobam os envolvidos no projeto, ou cujos interesses podem ser positiva ou negativamente afetados pela execução ou pelos resultados do projeto. Elas também podem exercer influência sobre o projeto e seus resultados [PMI 2012, p. 394].

Neste sentido, podemos distinguir dois tipos de *Stakeholders*:

- Os Membros de equipa (*Team Members*) – São colaboradores ou especialistas experientes e implicados no desenvolvimento detalhado do âmbito do projeto, podendo oferecer conhecimentos especializados na definição de atividades.
- Outros Stakeholders (*Other Stakeholders*) podem ser um indivíduo ou grupo de indivíduos ou organizações que podem ser afetados dependendo da decisão a tomar e da relação que têm com as atividades a ser desenvolvidas para atingir os objetivos definidos.

A equipa do projeto, inclui o gestor do projeto, a equipa de gestão do projeto, e outros membros da equipa que executam trabalho no projeto mas não estão necessariamente envolvidos na sua gestão [PMI 2012, p. 26].

Ou seja, a equipa do projeto é constituída por todos estes tipos de membros, *Project Manager*, *Team Member* e *Other Stakeholders* de forma a envolver todos os colaboradores no projeto e extrair todas as entidades importantes para a realização do projeto (Ver Figura 11). A equipa do projeto identifica interna e externamente todas as

conclusões aconselhadas a fim de determinar os requisitos do projeto e as expectativas de todas as partes envolvidas.

Relativamente à tomada de decisão sobre alterações ao projeto, é constituída uma equipa de análise que verifica todas as atividades críticas enumeradas pelos diversos intervenientes, e que de modo participativo e colaborativo (ex: através de votação) interagem de forma a assegurar uma linguagem comum e o sucesso do resultado partilhado por todos os intervenientes. O Project Manager neste sentido, deve moderar e intervir de acordo com a experiência e os objetivos a atingir.

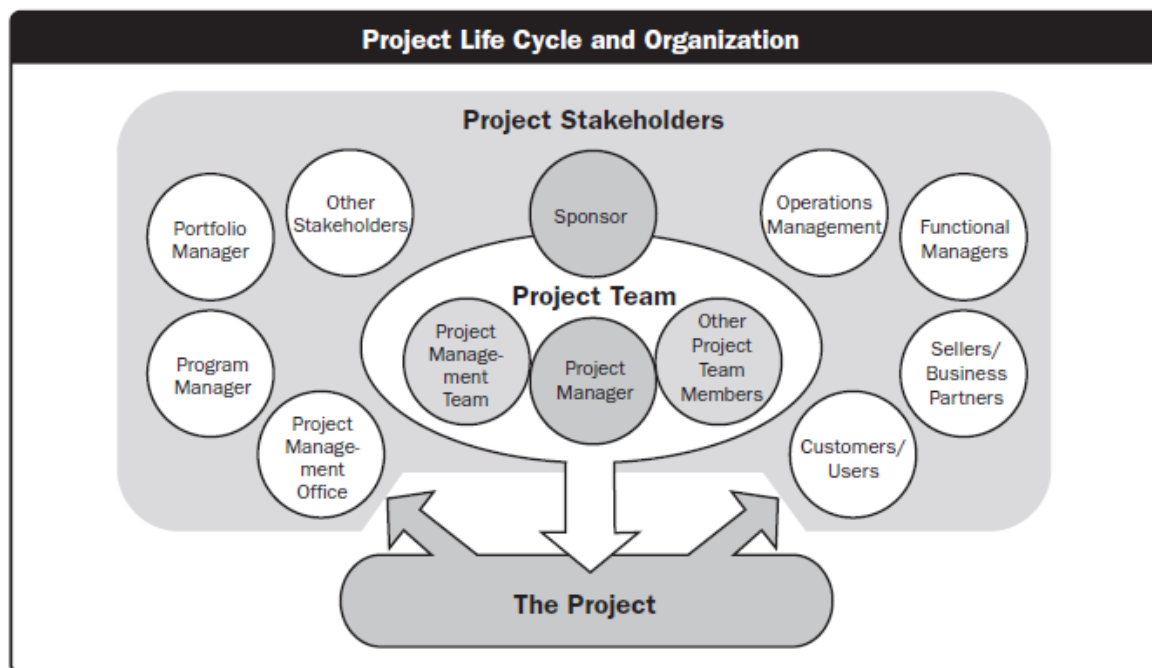


Figura 11 - Relação entre a Equipe de Projeto e outros Stakeholders (retirado de PMBOK guide, 2013).

### 3.4.7 FATORES CRÍTICOS DE SUCESSO E INSUCESSO NA GESTÃO DE PROJETOS

Vários são os aspetos cruciais para o sucesso na gestão do projeto conhecidos como Fatores Críticos de Sucesso. De uma forma geral, são elementos que determinam o melhor desempenho nas muitas atividades realizadas. Neste sentido os projetos devem estar dentro das restrições de âmbito, tempo, custo, qualidade, recursos e riscos



válidos devidamente definidos para poderem aspirar ao sucesso a que se propõem. A comunicação ao longo do projeto e a divisão de projetos complexos em partes menores deve ser uma realidade para um desfecho de sucesso.

No entanto, todos os aspetos devem ser equilibrados e devidamente medidos, para evitar casos de insucesso que resultam de falhas por deficiências ao nível de muitos destes aspetos enumerados. Por exemplo, a falta de clareza no que se vai fazer ou a preocupação excessiva com os tempos e custos podem corromper a falta de entendimento do âmbito. O crescimento do âmbito de uma forma incontrolada por solicitações por parte do cliente em relação a necessidades que não foram solicitadas nem orçamentadas podem provocar o mesmo desvio. São vários os fatores que podem influenciar negativamente um projeto, sobretudo a falta de comunicação, a ausência de um planeamento ou a pobreza do mesmo, por considerar-se que é perda de tempo, a falta de liderança, a inadequação de recursos afetados a determinado projeto podem provocar a falha de gestão do projeto.

### 3.5 CONCLUSÕES OU NOTAS FINAIS

Um Projeto é um esforço temporário empreendido para criar um produto tendo em vista a satisfação do cliente [PMI 2012, p. 3]. Gestão de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas adequadas às atividades do projeto, para satisfazer os seus requisitos. [PMI 2012, p. 5].

A gestão de Projetos caracteriza-se em dez áreas de conhecimento principais. São elas, gestão do âmbito, tempo, custo, qualidade, recursos humanos, comunicação, riscos, aquisições, integração e Stakeholders.

A aplicação dos conhecimentos requer a adoção eficaz de processos apropriados. Os cinco grupos de processos para a gestão de projetos são: iniciação, planeamento, execução, monitorização e controlo e encerramento ou finalização.

De acordo com o PMBOK [PMBOK Guide 2013], a gestão da qualidade do projeto engloba a gestão do projeto e do produto do projeto e aplica-se a todos os projetos, independentemente da natureza do produto. As medidas e técnicas de qualidade do produto são específicas do tipo de produto resultante do projeto. Por isso, o âmbito e objetivos do projeto devem ser bem-definidos.

O risco torna-se difícil de estimar quando os fatores de risco não são identificados no início. Embora o risco esteja em todas as fases do projeto, é preventivo quando identificado antes. De acordo com o PMBOK [PMI 2012, p. 226], “os objetivos da gestão dos riscos são aumentar a probabilidade de execução e reduzir a probabilidade dos mesmos no projeto”. Devem ser previstos para evitar situações desfavoráveis.

A comunicação é o fator chave para o sucesso de um projeto, onde participam diferentes intervenientes (interessados) com diferentes papéis (*Roles*), onde se destacam Gestor de projeto (*Project Manager*), Membro de equipa (*Team Member*) e outros Stakeholders (*Other Stakeholders*).

O Gestor de Projeto é a pessoa designada para a condução do projeto, com a missão de zelar para que os objetivos do projeto sejam atingidos. O gestor do projeto caracteriza-se por um perfil profissional com domínio e experiência especializados nos processos e nas áreas de conhecimento da gestão de projetos. A responsabilidade, ética, honestidade, liderança, autocontrolo e eficácia são alguns dos requisitos que um gestor de projeto deve possuir.

Os intervenientes ou partes interessadas (*Stakeholders*) são pessoas, grupos ou organizações que podem afetar, ser afetados ou sentirem-se afetados por uma decisão, atividade ou resultado de um projeto. São parte integrante no projeto, cujos interesses podem ser positiva ou negativamente afetados pela execução ou pelos resultados do projeto. Elas também podem exercer influência sobre o projeto e suas saídas. [PMI 2012, p. 394].

A equipa de projeto envolve todos os intervenientes que interagem com o projeto através da partilha de comunicação.

A equipa de análise é constituída por vários intervenientes, de diferentes perfis que tomam decisões tendo em conta a satisfação dos objetivos do cliente.

A determinação do sucesso ou do fracasso de um projeto exige o desenvolvimento de padrões de desempenho no projeto, os quais podem ser comparados aos resultados produzidos.



## 4. ANÁLISE DO PROBLEMA

### 4.1 INTRODUÇÃO

A área de gestão de projetos de software fortaleceu-se ao longo dos anos, motivada pela necessidade de garantir a qualidade e o sucesso dos projetos de software. Ao aliar conceitos clássicos da área de administração, tais como planeamento, coordenação, e controlo, com elementos específicos da área de software, a gestão de projetos de software configura-se como multidisciplinar e integradora de aspetos tanto estruturais como técnicos.

Por sua vez, o processo de desenvolvimento de software é formado por uma série de atividades técnicas e organizacionais que envolvem metodologias e ferramentas para elaborar, planear, controlar e executar as atividades do projeto de desenvolvimento de software. As atividades técnicas englobam desde as metodologias de desenvolvimento até à infraestrutura do ambiente.

As atividades organizacionais vão desde o atendimento às necessidades da organização até ao relacionamento com os recursos humanos e Stakeholders envolvidos em todo o ciclo de vida do projeto.

Diversos modelos de gestão são aplicados como referência nos mais diversos segmentos para garantirem o sucesso do projeto.

Neste projeto de mestrado foi proposta a modelação e desenvolvimento de uma aplicação web de suporte à gestão de projetos, que promova as melhores práticas de gestão de projetos de software. A aplicação deve fazer a gestão de diversos aspetos técnicos e de suporte ao projeto, incluindo a gestão de requisitos, elementos de modelo

tais como casos de uso e entidades, elementos de trabalho no projeto tais como tarefas, recursos humanos, bem como a divulgação e distribuição de toda a informação privilegiando a comunicação comum entre todos os intervenientes.

De acordo com o modelo PMBOK, é importante que a aplicação consolide todas as áreas de conhecimento, onde cada colaborador do projeto assuma um papel interventivo através do seu contributo na transmissão dos seus conhecimentos e consequentemente toda a informação recolhida seja valorizada. Neste sentido, a aplicação deve permitir gerir todos os recursos que dele farão parte distinguindo requisitos, *Use Cases* e tarefas.

## 4.2 REQUISITOS

A construção de um produto de software envolve o desenrolar de um projeto que consiste na união de várias fases de desenvolvimento, não apenas técnicas, mas de planeamento e estratégia. Saber o que é fundamental, baseado nos problemas que o sistema precisa resolver, pode fazer a diferença entre um projeto de sucesso e outro que resulta em fracasso.

Neste sentido este projeto pretende desenvolver uma aplicação web usando Spring MVC para a gestão de projetos de desenvolvimento de software. Com este objetivo e tendo em conta o aglomerado de funcionalidades que dão suporte à gestão de projetos baseado no modelo PMBOK foram definidos os seguintes requisitos:

- Desenvolvimento de uma aplicação que corra em ambiente Web (browser), e permita:

- ❖ Criar, Listar e editar Projetos de Software.
- ❖ Para cada Projeto, permita definir/editar dados do projeto como por exemplo, o âmbito do projeto, a data início e fim, etc.
- ❖ Associado a cada projeto, permita criar e manter uma lista de Stakeholders;
- ❖ Associado a cada projeto, permita criar e manter uma lista de membros da equipa de projeto;
- ❖ Associado a cada projeto, permita criar e manter uma lista de Stakeholders (formada por alguns de entre: gestor de projeto, membros de equipa ou outros Stakeholders), que formam o grupo de análise de pedidos de alteração ao projeto;
- ❖ O sistema deve ter proteção por login password e possuir perfis diferenciados para gestor de projeto; membros da equipa e outros Stakeholders;
- ❖ Um utilizador pode ser gestor de projeto num dado projeto A, mas ser apenas membro de equipa (não gestor) num outro projeto B, e ainda ser “Outro Stakeholder” num terceiro projeto C. As suas permissões deverão ser diferentes para cada um desses projetos.
- ❖ Associado a cada projeto, permita criar e manter uma lista de Requisitos;
- ❖ Cada Requisito pode estar no estado: CRIADO (inicial), ANULADO, EM\_DESENVOLVIMENTO, EM\_TESTES, ALTERADO;

- ❖ Deve ser mantido um histórico de alterações de estado por requisito;
- ❖ Associado a cada projeto, permita criar e manter uma lista de Milestones (datas marco e respectivas descrições);
- ❖ Associado a cada projeto, permita criar e manter uma lista de casos de uso, uma lista de Entidades de Domínio, e uma lista de Ecrãs do sistema.
- ❖ A cada requisito pode ser associado um ou mais Stakeholders proponentes;
- ❖ A cada Stakeholder proponente pode ser associado um ou mais requisitos;
- ❖ A cada caso de uso pode ser associado um ou mais membros da equipa;
- ❖ A cada membro da equipa pode ser associado um ou mais casos de uso;
- ❖ Cada caso de uso deve ter um nome, uma Descrição, o tempo estimado e real para o seu “desenvolvimento”, e datas início e fim estimadas e reais para desenvolvimento;
- ❖ A cada caso de uso pode ser associado um ou mais requisitos;
- ❖ A cada requisito pode ser associado um ou mais casos de uso;
- ❖ A cada caso de uso pode ser associada uma ou mais entidades do domínio;
- ❖ A cada entidade do domínio pode ser associado um ou mais casos de uso;



- ❖ A cada caso de uso pode ser associado um ou mais ecrãs;
- ❖ A cada ecrã pode ser associado um ou mais casos de uso;
- O sistema deve gerar matrizes de rastreabilidade que relacionem:
  - ❖ Casos de uso e Requisitos;
  - ❖ Casos de Uso e Entidades;
  - ❖ Casos de Uso e Ecrãs;
  - ❖ Casos de Uso e Stakeholders proponentes;
  - ❖ Casos de Uso e membros da equipa responsáveis;
- O sistema deve ainda permitir a seguinte funcionalidade de gestão de alterações:
  - ❖ Listagem e Criação de pedidos de alteração de âmbito, custo (esforço), calendário;
  - ❖ Um pedido de alteração deve identificar os requisitos iniciais afetados;
  - ❖ Um pedido de alteração pode estar no estado: PEDIDO, VOTAÇÃO; APROVADO, REJEITADO, CANCELADO; CONCLUÍDO.
  - ❖ Deve ser mantido um histórico de alterações de estado por pedido de alteração;
  - ❖ Um pedido de alteração deve ser analisado pelo grupo de análise de alterações (conjunto de Stakeholders), que o pode aprovar ou rejeitar;

- O sistema deve gerar relatórios (ecrã e pdf):
  - ❖ Cada uma das listas e matrizes acima;
  - ❖ Lista de “itens” afetados pela alteração de outro “item” (ex.: Todos os ecrãs afetados pela alteração de um caso de uso; Todos os Ecrãs afetados pela alteração de uma entidade do domínio; Todos os Casos de Uso afetados pela alteração de um ecrã; ...).
  - ❖ Lista de Requisitos do projeto (num dado estado...).
  - ❖ Lista de pedidos de alteração (num dado estado: aprovados, rejeitados,...).
- Estudar a possibilidade/facilidade de o sistema integrar com uma ferramenta de modelação UML, como por ex. o StarUML; e uma ferramenta de sketching de ecrãs.

A definição de requisitos é uma tarefa árdua e difícil de ser estabilizada, pois à medida que são desenvolvidas ou detalhadas as necessidades do sistema vão aparecendo outras necessidades que provocam a retroação do desenho de sistema e portanto a reengenharia de requisitos. Neste projeto, após a proposta e definição do objetivo, foram definidos os requisitos anteriormente definidos os quais foram objeto de análise e modelação e apenas alguns implementados por falta de tempo ficando os restantes para desenvolvimento futuro.

### 4.3 ESTRUTURAÇÃO DOS REQUISITOS

Com a definição de requisitos estabilizada e bem definida, é necessário estruturar os requisitos, de forma a organizá-los de acordo com a sua envolvimento ou domínio para permitir o seu relacionamento bem como definir a lógica de negócio que os irá implementar. De acordo com o modelo PMBOK é importante definir uma Estrutura analítica do projeto/ Work Breakdown Structure (EAP/WBS), de forma a definir hierarquias de decomposição de requisitos.

Os processos definidos no PMBOK são apresentados de forma sequencial a fim de permitir que os resultados de um processo sejam entrada para outros processos, embora muitos na utilização prática ocorram em paralelo.

Face ao exposto é fácil compreender a transferência do conhecimento definido no modelo seguido. O desenvolvimento do EAP/WBS é um trabalho complexo e que exige tempo e esforço, mas que assegura a estabilidade de processos do sistema de acordo com as necessidades definidas pelo cliente. Para além disso a criação e manutenção da EAP/WBS do projeto é um processo de refinamento contínuo e que por isso deve acompanhar as alterações que, no âmbito do processo de gestão de alterações, forem sendo introduzidas no projeto.

Foram estudados analiticamente todos os requisitos para o projeto de forma a perceber quais as funcionalidades e de que maneira se relacionam para produzir um sistema estruturado, com capacidade de ser adaptado a novas necessidades no futuro e que permita sobretudo dar suporte ao modelo PMBOK.

Para melhor compreender o arranjo do sistema, procedeu-se à enumeração, definição e análise de requisitos da ferramenta os quais foram organizados em grupos e na seguinte ordem de prioridades de desenvolvimento:

1. Implementação de gestão de utilizadores
2. Implementação de gestão de Stakeholders
3. Implementação de gestão de habilidades (*Skills*)
4. Implementação de gestão de equipas
5. Implementação de gestão de pedidos
6. Implementação de gestão de Entidades de domínio
7. Implementação de gestão atividades
8. Implementação de gestão de diagramas
9. Implementação de gestão de históricos
10. Implementação de gestão de votações
11. Implementação de gestão de projetos
12. Implementação de ecrãs de sistemas

Recorrendo a um diagrama de pacotes ou implementação (Figura 12) segue uma representação gráfica de forma a melhor compreender a relação entre os diferentes grupos de requisitos a serem implementados.

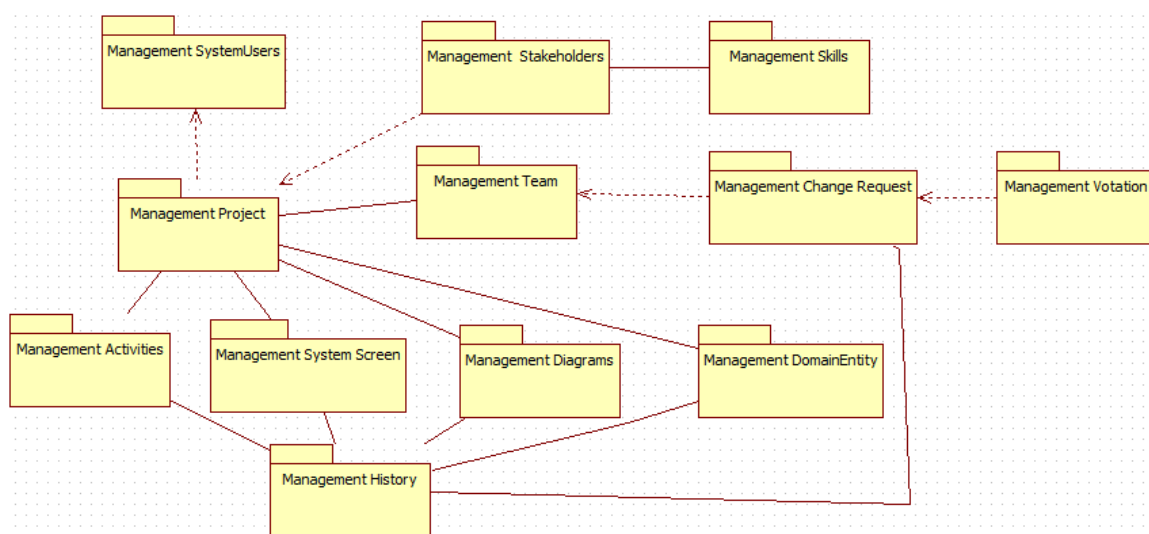


Figura 12 - Diagrama de Pacotes da aplicação.

De acordo com o PMBOK, após definidos os objetivos do projeto, foi preciso definir as diferentes áreas de conhecimento que se distribuem nos diferentes pacotes acima definidos.

Foi possível definir um prazo que será apresentado mais à frente num cronograma. Como houve apenas um envolvido no desenvolvimento deste projeto e a efetuar todo este estudo, o autor deste relatório, a gestão de riscos, de aquisição e integração não foram alvos de ênfase principal, pese embora o desenvolvimento da aplicação conjugue todas essas etapas.

A gestão da comunicação e de Stakeholders foram enfatizadas neste desenvolvimento, com a construção a modelação definida em etapas de desenvolvimento tendo por base a integração cada vez mais refinada de cada parte de desenvolvimento.

Neste contexto, seguindo uma estrutura hierárquica sequencial, foi necessário implementar uma gestão de utilizadores que permite distinguir diferentes perfis, controlando o seu acesso ao sistema.

Definidos os objetivos principais que compõem o âmbito do projeto, é necessário perceber que comportamento terá o sistema e qual a sequência da informação a gerir em cada processo. Cada operação deverá permitir uma sequencialidade de tarefas, permitindo assim ao gestor saber quais são as tarefas prioritárias a desenvolver e a sua relação no contexto processual. A aplicação deve permitir a aprendizagem ao longo da realização de vários projetos, permitindo melhorar a sua performance e obter valor acrescentado em cada projeto que se desenvolve. Neste sentido a aplicação deve suportar uma estrutura que permita armazenar o histórico das alterações aos vários

elementos do projeto (requisitos, casos de uso, entidades, etc.), e permita a rastreabilidade entre elementos e fases de cada projeto.

Com a atribuição e distribuição de Stakeholders, a aplicação deve permitir a criação de equipas de projeto bem como uma equipa de análise por projeto, que assegurará a análise crítica de cada pedido de alteração ao projeto.

No sentido de favorecer uma comunicação comum onde todos os intervenientes/Stakeholders participam, a equipa de análise deverá ter ao seu dispor uma ferramenta que permita ultrapassar resistências, valorizar a responsabilidade de cada interveniente e favorecer a sua intervenção através de um sistema de votação com possibilidade de justificação de voto.

De acordo com o modelo definido, cada funcionalidade tem uma importância relevante na arquitetura do sistema tendo em vista a sua implementação.

A gestão de projetos e a gestão de pedidos de alteração são pacotes de gestão compostos por outras componentes de gestão específicos intrínsecos à lógica de negócio do sistema em desenvolvimento.

O sistema de gestão de utilizadores deve permitir o controlo do acesso às funcionalidades do sistema através de login e password. Cada utilizador do sistema deve ser classificado pelo seu tipo podendo ser Administrador ou apenas Utilizador do sistema. Um utilizador que tenha o perfil de Administrador deverá permitir para além de todas as funcionalidades que um utilizador simples acumula, a responsabilidade de gerir utilizadores e seus perfis no Sistema.

A gestão de Stakeholders, deve assegurar a criação de diferentes perfis no contexto do projeto a desenvolver. Neste sentido, cada utilizador do sistema, independentemente do perfil que tenha ao nível do sistema, deve também ser-lhe

associado um perfil/papel ao nível da lógica do negócio, num dado projeto. Neste caso, cada utilizador é catalogado como “colaborador” e categorizado ao nível de um projeto como: *Project Manager*, *Team Member* e *Other Stakeholder*.

O papel de *Project Manager* permite ao colaborador ser soberano no projeto em desenvolvimento, podendo controlar todas as etapas e gerir todas as decisões ao nível do projeto. Cabe ao Project Manager criar o projeto de que ele é líder, criar a equipa de projeto e a equipa de análise, bem como monitorizar todas as etapas de cada requisito a ser desenvolvido, as tarefas, *Use Cases* Associados, Entidades de Domínio e Ecrãs de Sistema associados.

*Team Member* é outro papel associado na lógica de negócio atribuída a um colaborador envolvido num projeto. Este papel confere ao colaborador, fazer parte do projeto, criar casos de uso, requisitos, tarefas, propor alterações de âmbito ou calendarização, criar entidades de domínio e/ou ecrãs de sistema. Pode ser-lhe atribuída a responsabilidade de votar, se este fizer parte da equipa de análise, podendo nesse caso acordar ou rejeitar alterações propostas pelos vários Stakeholders do projeto.

*Other Stakeholders* é o papel com que uma pessoa interessada no projeto é rotulada, sendo restringida a sua responsabilidade à criação de requisitos, propor alterações através da elaboração de um pedido de alteração e integrar a lista de votantes no caso de ser-lhe concedida a responsabilidade de votar pelo Project Manager.

A gestão de habilidades (*skills*), permite qualificar cada colaborador do projeto no sistema em desenvolvimento. Nela recai a responsabilidade de gerir quais as habilidades que cada colaborador tem em cada projeto, sendo determinantes para a escolha de formação de equipas de análise e seleção de informação por parte do Project Manager.

Neste contexto um utilizador pode ter diferentes intervenções em diferentes projetos, podendo variar as suas habilidades de acordo com o papel atribuído. Dentro dos mais usuais um colaborador pode ser analista num projeto e programador noutro projeto ou até mesmo acumular mais do que uma habilidade noutro projeto completamente diferente.

A gestão de equipas é responsável pela definição e gestão de equipas no projeto. Neste sentido o sistema deve permitir ao Project Manager criar a equipa de projeto e a equipa de análise que é constituída pelo Project Manager e por alguns elementos da equipa do projeto nomeados também pelo Project Manager (Ver Figura 13).

Considera-se a equipa de projeto todos os *Team Members* do projeto, seja ele qual for o seu papel. A equipa de projeto deve assegurar e privilegiar a comunicação comum de forma a identificar e refinar os requisitos necessários ao desenvolvimento do projeto.

Por sua vez, a equipa de análise do projeto terá a seu cargo a responsabilidade de gerir sobretudo situações de requisitos conflituosos e analisar pedidos de alteração de requisitos do projeto, bem como neutralizar e clarificar todos os aspetos de difícil ou complexa compreensão.

A gestão de pedidos de alteração é responsável pelas solicitações dos intervenientes do projeto ao nível dos requisitos. Neste sentido todo o *Stakeholder* (*Team Member* ou outros interessados (Other Stakeholders) ou envolvido no projeto) tem o direito de solicitar qualquer alteração a fazer ao nível dos requisitos. Neste sentido a gestão de requisitos deve sempre privilegiar a congruência de todos os requisitos tendo em conta os objetivos do projeto e a comunicação aberta e partilhada por todos os intervenientes. A equipa de análise assume um papel preponderante na



gestão de alterações já que cabe a cada elemento que a compõe a responsabilidade de dar o seu ponto de vista de uma forma justificada.

A funcionalidade de gestão de pedidos de alteração colabora diretamente com outras componentes de sistema, nomeadamente a gestão de votação e de projeto, que são pilares chaves para a coordenação, rastreabilidade e execução do projeto.

A gestão de votação é responsável pela gestão de votações dos diferentes intervenientes que constituem a equipa de análise para cada pedido de alteração.

Um pedido de alteração pode solicitar vários tipos de modificação ao plano de um projeto, tais como, alterações de âmbito, custo ou calendário. Deve ser mantido a este nível um histórico de alterações que terão necessariamente um estado, podendo este variar como, Pedido, Votação, Aprovado, Rejeitado, Cancelado e Concluído. Dependendo do ciclo de vida de estado. Os intervenientes que fazem parte da equipa de análise serão chamados a intervir de forma a aprovar ou rejeitar as alterações solicitadas.

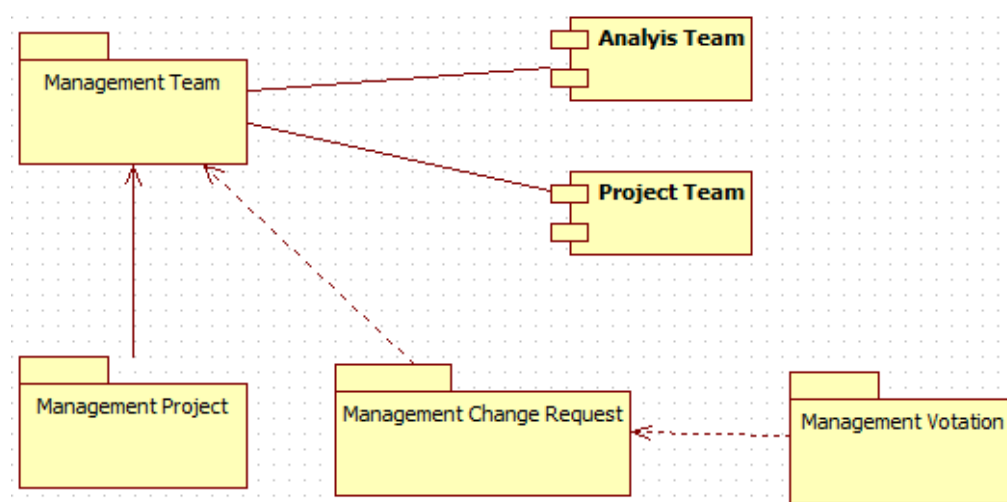


Figura 13 - Composição do Pacote de Gestão de Equipa.

A gestão de Entidades de domínio é uma funcionalidade que foi modelada mas não está implementada. Esta funcionalidade é responsável pela criação de Entidades de Domínio no projeto e a sua associação a casos de uso. Neste sentido todos os *Team Members* de um projeto devem poder criar entidades de domínio ou pelo menos propor diferentes entidades de domínio para fazerem parte do projeto a desenvolver. A gestão de Entidades de Domínio tem uma relação direta com outras componentes do sistema, tais como gestão de projetos e gestão de atividades.

A gestão de Ecrãs de Sistema, foi modelada mas não implementada. A mesma é responsável pela criação de ecrãs ou mockups para cada caso de uso do projeto a ser desenvolvido. Neste sentido, esta componente permite de imediato apresentar um ecrã de sistema potencial por caso de uso do projeto.

De acordo com o modelo PMBOK é importante ter presente os outputs a disponibilizar ao utilizador resultante da realização de diferentes operações. Por este motivo e por razões de estimativa de custo é necessário ter precocemente uma identificação de todos os outputs a disponibilizar pelo sistema.

Esta componente terá relação com outras componentes, nomeadamente a gestão de projeto onde são definidos níveis de rastreabilidade importantes para a integração do sistema como um todo.

A gestão de atividades associadas a um caso de uso foi modelada mas não implementada. Esta funcionalidade é responsável pelo estabelecimento de sequenciação de atividades por cada caso de uso. Neste sentido quando um caso de uso é definido, deve ser definida toda a descrição de funcionamento, incluindo pressupostos, inputs e outputs. É importante transportar todas as atividades envolvidas para o sistema a que nada seja deixado ao acaso, evitando que a orçamentação e o

timing não tenham desvios maiores. A componente de gestão de atividades está intimamente relacionada com outras componentes, nomeadamente, a gestão de projeto e de casos de uso.

A gestão de diagramas foi modelada e não implementada e é responsável pela gestão de todos os diagramas a serem desenvolvidos no projeto. Neste sentido, será possível importar ficheiros derivados de outras ferramentas que poderão fazer parte ou não do pacote de instalação, dependendo da inclusão de licenças de utilização. Segundo o PMBOK, um sistema de gestão deve reunir todas as melhores práticas de gestão ao nível dos recursos. Hoje em dia as ferramentas Open Source assistem a uma comunidade cada vez mais interventiva e colaborativa disposta a participar com o melhor que é feito, mesmo que seja apenas uma parte da solução. Neste sentido, a gestão de diagramas deve permitir a importação de ficheiros de ferramentas como o StarUml para permitir uma análise visual de todos os casos de uso e entidades do domínio no contexto do projeto em desenvolvimento.

A gestão de históricos foi uma funcionalidade bastante explorada ao nível da modelação não sendo possível a sua implementação por razões de tempo de implementação. Esta funcionalidade será responsável pela gestão do histórico de todo o sistema. Neste sentido, toda a aplicação tem um valor acrescentado com esta implementação. É responsável pela dinâmica dos projetos a longo prazo, permitindo extrair em cada novo projeto a informação nas diferentes áreas de conhecimentos referidas no modelo PMBOK. A gestão de históricos tem uma relação direta com outras componentes nomeadamente a gestão de projetos, gestão de pedidos de alteração, gestão de entidades de domínio e gestão de ecrãs.

A gestão de projetos, permite essencialmente criar um projeto, onde são disponibilizadas todas as funcionalidades de acordo com os objetivos/âmbito traçado no projeto. Deve ser permitida a criação de projetos a todos os utilizadores do sistema assumindo este, neste caso, o papel de Project Manager (Ver Figura 14).

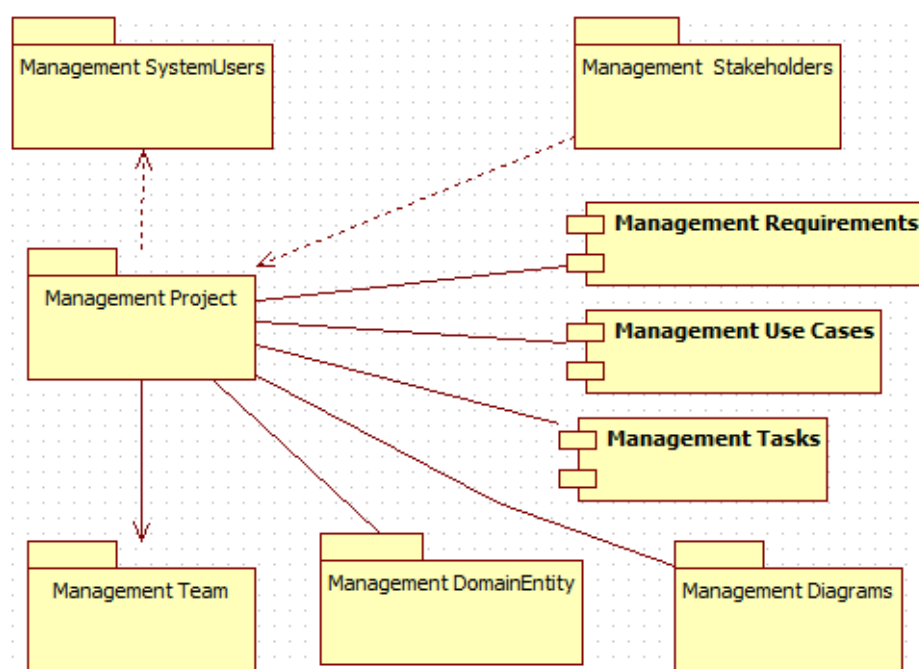


Figura 14 - Composição do Pacote de Gestão de Projeto.

A gestão de projetos deve ser o centro de todo o sistema a contactar todas as componentes de gestão que compõem o sistema total. Neste sentido a gestão de projetos deve permitir a criação de diferentes colaboradores, com diferentes habilidades (*Skills*) em cada projeto. Dependendo das suas habilidades o colaborador deve ter no sistema o grau de intervenção apropriado em cada projeto.

Cada projeto deverá suportar um conjunto de requisitos, permitindo a sua gestão.

Cada requisito pode estar no estado: Criado, Anulado, Em Votação, Em desenvolvimento, Em testes e Alterado.

Deverá ser mantido o histórico de alterações de estado por requisito.

Cada Projeto deve manter uma lista de “Milestones” tais como datas marco e respectivas descrições.

Cada Projeto deve permitir criar e manter uma lista de casos de uso, uma lista de Entidades de Domínio bem como uma lista de Ecrãs.

É neste contexto e com a necessidade acrescentada de haver uma ferramenta que dê suporte a estas funcionalidades latentes que foi modelada e desenvolvida uma aplicação web que sustenta este projeto.

#### 4.4 CASOS DE USO

Para representar e melhor perceber o levantamento de requisitos elaborados, bem como os intervenientes que interagem com o sistema, de forma a garantir que o sistema desenvolvido está de acordo com as necessidades definidas, foram desenvolvidos diagramas de “*Use Cases*”.

Como foi referido anteriormente, foi preciso distinguir as funcionalidades por perfil de utilizador, neste caso por tipo de interveniente. Ou seja, de forma a garantir a comunicação comum, foram analisados e definidos os modelos de casos de uso na ótica do utilizador com o papel de *Project Manager*, *Team Member* e *Other Stakeholder*.

Como descrito anteriormente, o perfil *Project Manager* assume uma responsabilidade de gestor de todas as tarefas dentro do projeto. Neste caso recai sobre ele a responsabilidade que um verdadeiro *Project Manager* detém. Ele é o líder da equipa, que gere as despesas do projeto bem como as decisões de grande relevo.

O perfil *Team Member* representa um *Stakeholder* membro da equipa de projeto e que colabora diariamente no projeto. Este pode, por exemplo, criar requisitos, casos

de uso e tarefas. Este tipo de perfil representa todos membros da equipa, que têm responsabilidade de gerir o seu trabalho, possuindo também conhecimentos na área em que trabalham.

Por fim, *Other Stakeholders* é o tipo de *Stakeholder* que representa aqueles intervenientes que poderão dar informação importante ao desenvolvimento do projeto do ponto de vista do cliente ou utilizador final. De acordo com o modelo PMBOK, muitas vezes estes são intervenientes chave para o desenvolvimento mas que não lhes é dada a importância devida. Neste papel pode encaixar perfeitamente um elemento de responsabilidade como por exemplo um Presidente de Câmara ou chefe de apoio ao desenvolvimento por parte do cliente, dependendo de cada projeto específico.

Nos diagramas de casos de uso adiante apresentados (Figuras 15 a 18) serão descritas as funcionalidades principais de cada perfil no projeto. Toda a arquitetura do sistema foi pensada de forma integrada, minimizando possíveis erros de redundância e incongruência nos diferentes fluxos de informação. Para facilitar a compreensão do modelo, e de forma a desburocratizar todas as possíveis funcionalidades que o utilizador poderá ter em cada perfil, é apresentado também um diagrama de casos de uso para os perfis de utilizador. Neste caso para o perfil de administrador.

O diagrama de casos de uso apresentado na Figura 15 representa as funcionalidades modeladas no sistema para o ator *Project Manager*. A maior parte destas funcionalidades foi já implementada na aplicação desenvolvida. Na figura, todos os *Use Cases* de cor mais clara (amarela) foram desenvolvidos na aplicação. As outras funcionalidades de cor diferente não foram ainda implementadas devido a restrições de tempo, ficando para implementação futura.

No diagrama podemos verificar que o *Project Manager* é um perfil de utilizador na lógica de negócio, que apenas faz sentido quando o utilizador é reconhecido pelo sistema, ou seja, que é autenticado pelo sistema. Daí a relação de generalização entre o ator “*System User*” e “*Project Manager*”. Neste sentido, o utilizador contacta com o sistema fazendo inicialmente a sua obrigatória autenticação com username e password. De acordo com a sua autenticação, o perfil “*Project Manager*” é atribuído, em cada projeto, ao *Stakeholder* que é gestor de projeto.

O *Project Manager*, poderá criar projetos e gerir todas as funcionalidades que fazem parte do processo de gestão de um projeto. Sempre que é criado um projeto, é “gerado” automaticamente um perfil *Project Manager* para o criador, permitindo-lhe aceder a todas outras funcionalidades que um “*Project Manager*” tem no seu perfil.

Depois de criado o projeto, o *Project Manager* pode criar a equipa de projeto, associando ou criando *Stakeholders* e *Team Members* à sua equipa de projeto. Note-se que um *Team Member* é um *Stakeholder* colaborador ativo no projeto, com habilidades específicas relacionadas com o desenvolvimento do projeto, tendo pelo menos uma habilidade (*Skill*) profissional. O *Stakeholder* é um interveniente que pode ser relevante ao desenvolvimento do projeto, mas que pode não ter conhecimentos na área técnica de desenvolvimento de software, mas pode ser especialista no domínio de aplicação do software em desenvolvimento.





Para além de construir a equipa de projeto, o *Project Manager* pode criar a equipa de análise que assume um importante papel na tomada de decisão em propostas feitas por qualquer membro da equipa, mantendo a comunicação comum e reduzindo o conflito entre os intervenientes. Cabe ao *Project Manager* a responsabilidade de nomear entre os elementos da equipa aqueles que farão parte da equipa de análise, podendo esta ser constituída pelos dois tipos de utilizadores.

Com a finalidade de monitorizar os requisitos do sistema e quais as tarefas que os irão implementar, o *Project Manager* pode registar ou alterar requisitos e definir e gerir tarefas a ele relacionadas. De acordo com a sua sequencialidade é possível definir a sua precedência. No seguimento do modelo PMBOK todos os requisitos podem sofrer alterações sendo refinados até à sua estabilização, por todos os intervenientes do projeto.

Para monitorizar e melhor compreender a relação entre os intervenientes do projeto em desenvolvimento, é disponibilizada a funcionalidade de criação de *Use Cases*. Os *Use Cases* são uma forma de perceber quais os intervenientes e de que forma se relacionam com as diferentes entidades de Domínio. Neste sentido, os *Use Cases* permitem associar requisitos, tarefas, criar ou associar entidades do domínio, ecrãs de sistema, modelos de sistema, bem como criação de atividades.

No domínio de um projeto de software deve ter-se em consideração, quais as técnicas e melhores práticas utilizadas no desenvolvimento do projeto. Assim as atividades são importantes para permitir criar sequenciação de todas as atividades do *use case*, permitindo descrever o seu detalhe e precondições se necessário. As entidades de domínio representam todas as entidades envolvidas no domínio do projeto, relegando para os modelos do sistema através do suporte dos diagramas a sua

esquematização. Neste aspeto foi estudada a possibilidade de outras ferramentas como o StarUml integrarem o pacote de instalação da aplicação ou simplesmente integrar os ficheiros que deem suporte ao modelo de domínio. Os ecrãs de sistema permitem monitorizar todos os ecrãs do projeto. Neste sentido é possível ter uma lista prévia das mockups a serem implementadas para darem suporte ao desenvolvimento gráfico do projeto.

De acordo com o modelo PMBOK, é importante gerir os recursos disponíveis e disponibilizar de uma forma sustentada versões do projeto em desenvolvimento. Tendo em vista a gestão da integração e recorrendo a modelos ágeis, a disponibilização de todos os modelos iterativos no domínio do sistema é imprescindível para o sucesso do desenvolvimento do projeto. Neste sentido a criação de *use cases* assume um papel preponderante no desenrolar do projeto e permitindo a integração de todos os intervenientes e modelos do sistema.

Tendo em vista as alterações a que o sistema poderá ser submetido, qualquer um dos intervenientes da equipa de projeto pode sugerir alterações a requisitos ou ao âmbito do projeto, devendo o *Project Manager* solicitar a opinião dos membros da equipa de análise.

De forma democrática todos os intervenientes que compõem a equipa de análise participam com a sua opinião através de uma votação a qual é presidida pelo *Project Manager*. De acordo com a votação o *Project Manager* decide que resolução irá ter cada pedido de alteração solicitado.

No PMBOK a gestão de recursos humanos é um pilar das áreas de conhecimento que deve ser gerido com experiência, para evitar conflitos entre os intervenientes e fazer funcionar o grupo sem excessos de confiança e individualismo. O objetivo deve ser

obtido de acordo com a colaboração de todos numa gestão de risco calculada com o mínimo custo de qualidade. Por isso o sistema disponibiliza de uma forma interativa e apelativa a colaboração autónoma e livre de todos os intervenientes onde o *Project Manager* é o líder das responsabilidades a tomar.

O *Project Manager* tem ao longo do projeto uma visão superior e pode monitorizar passo a passo tudo que cada *Stakeholder* faz no projeto, por isso a configuração visual da aplicação também é distinguida ao nível de operações como a alteração de pedidos e votações.

O *Team Member* é outro perfil que a lógica de negócio do sistema disponibiliza e distingue no seio do desenvolvimento de um projeto. Como já foi referido, todo o utilizador pode assumir um dos três perfis no domínio do projeto depois de ser autenticado. O perfil *Team Member* permite determinadas funcionalidades, depois de ser adicionado pelo *Project Manager* no projeto. A Figura 16 ilustra o diagrama de casos de uso para o ator *Team Member*.

Assim *Team Member* é um utilizador do sistema que pode criar requisitos, casos de uso, tarefas, bem como proceder à sua alteração no caso de ser o autor da sua criação.

Com o conhecimento especializado no domínio de intervenção o *Team Member* tem associado uma ou várias habilidades (*Skills*) e pode criar *Use Cases* bem como associar tarefas e requisitos aos referidos *Use Cases*.

Toda a envolvência que o *Use Case* engloba é passível de ser gerida pelo *Team Member* que integra essa responsabilidade. Neste sentido pode criar modelos de sistema e os diagramas que o suportam, entidades de domínio e ecrãs do sistema.

Com conhecimentos técnicos e à medida que vai compreendendo cada requisito e o seu processo pode definir a precedência das atividades de cada *Use Case* bem como a sequenciação das tarefas no âmbito do projeto.

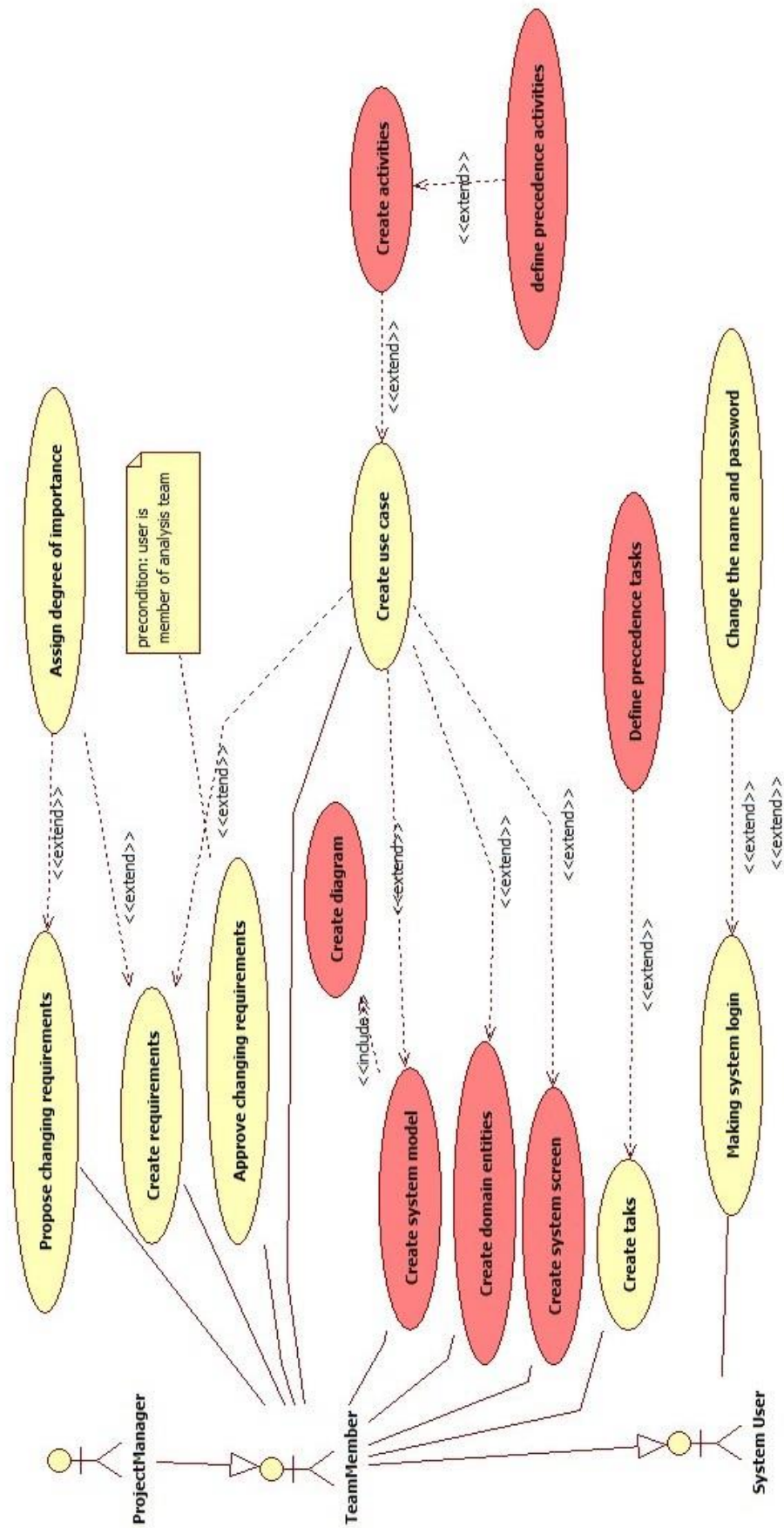


Figura 16 - Diagrama de Casos de uso do Team Member.

É importante perceber que um *Team Member* pode solicitar alterações a requisitos de acordo com o refinamento das tarefas a serem executadas a cada momento. No entanto nem todos os *Team Members* terão a oportunidade de pertencer à equipa de análise do projeto. Esse é uma responsabilidade do *Project Manager* que tem uma supervisão e constitui a equipa de análise de acordo com as necessidades do projeto e o seu conhecimento dos recursos disponíveis.

Quando o *Team Member* integra a equipa de análise, tem a responsabilidade de contribuir com a sua visão e experiência no desenvolvimento do projeto. Assim pode pronunciar-se interferindo nos pedidos de alteração solicitados por qualquer membro da equipa de projeto.

Na ótica do *Team Member*, a nível aplicacional, o sistema permite ao utilizador seguir o estado de cada alteração de requisito não tendo acesso geral a todas as tomadas de decisão relativas ao pedido de alteração solicitado.

De acordo com o modelo PMBOK, todos os intervenientes com responsabilidade devem intervir de acordo com a sua experiência e conhecimento, sem influências, de forma a não afetar o objetivo do projeto. Para evitar situações de conflito e promover a autonomia proposta pelo modelo, todos os *Stakeholders* têm um papel preponderante no desenvolvimento do projeto. Neste sentido a aplicação importa a gestão de *Stakeholders* fornecendo responsabilidades identificadas por utilizador e permitindo acompanhar o desenvolvimento do *Team Member* envolvido em cada requisito.

Com a gestão de *Stakeholders* como leme importante na gestão de projetos de software, todos os tipos de interveniente têm um papel importante no processo de gestão de projetos. Neste sentido foram analisadas e implementadas as funcionalidades

relativas ao perfil *Other Stakeholders*. Na Figura 17, o diagrama de casos de uso apresentado, representa as funcionalidades que um *Stakeholder* com este perfil pode executar.

No diagrama de casos de uso, *Other Stakeholder* é apresentado como “*Stakeholder Proponente*” de forma a permitir compreender melhor a contextualização do modelo de negócio do sistema com o modelo PMBOK que tem sido seguido.

À semelhança dos outros dois perfis que já foram apresentados na lógica de negócio, o “*Stakeholder Proponente*”, depois de ser autenticado pelo sistema e associado pelo *Project Manager* à equipa do projeto, tem um papel importante no desenrolar do projeto. O seu papel depende do seu conhecimento na área de intervenção relativamente ao objetivo pretendido. Como muitas vezes sabem o que querem mas não dominando os termos técnicos na área de software, não se conseguem expressar, assumem um papel de intervenção e não uma habilidade profissional relativa ao desenvolvimento do projeto.

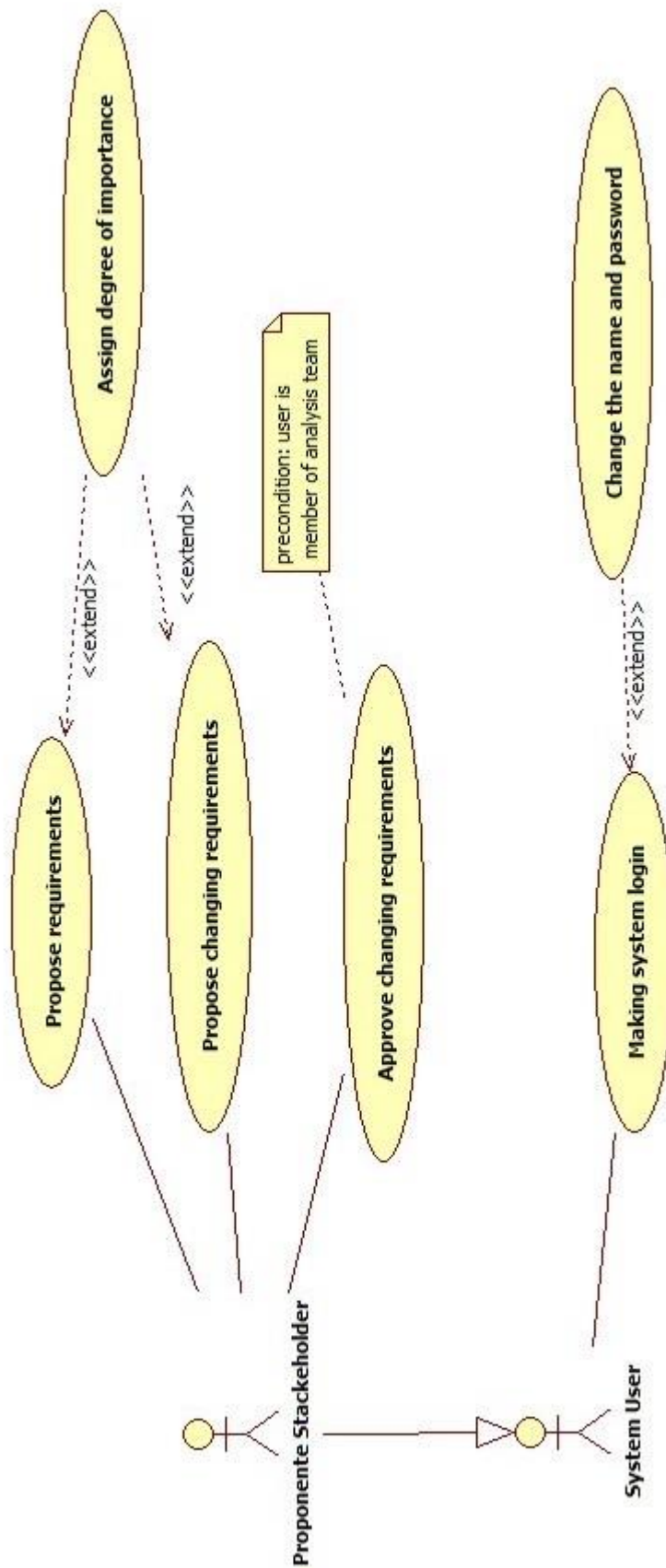


Figura 17 - Diagrama de casos de uso do Stakeholder proponente.



Como funções principais, o *“Stakeholder Proponente”* pode propor requisitos que achar convenientes na definição do projeto. Esta definição deve merecer o interesse relevante de toda a equipa de análise que avaliará a relevância da informação e o grau de interesse no projeto. A este ponto trata-se portanto de garantir a gestão de requisitos de que fala o modelo PMBOK.

No seguimento dos outros perfis na ótica do negócio do sistema, o *“Stakeholder Proponente”* pode ainda intervir nos pedidos de alteração dos requisitos ou do âmbito do projeto. Com efeito, esta funcionalidade depende da associação do *Stakeholder* em causa na equipa de análise. Como tem sido descrito, a equipa de análise liderada pelo *Project Manager*, permite a cada membro interferir diretamente na análise de alterações aos requisitos. Nestas condições o *“Stakeholder Proponente”* pode analisar pedidos de alteração aos requisitos ou âmbito do projeto, bem como participar na decisão de aceitação/rejeição desses pedidos, através de votação.

O modelo PMBOK garante assim o alinhamento da informação de todos os requisitos e a novidade participativa de outros intervenientes na definição e desenvolvimento de um sistema de gestão.

Na parte aplicativa, esta funcionalidade foi implementada, permitindo ao utilizador que cumpra estas condições poder, através de votação democrática, decidir de acordo com a sua experiência ou conhecimento do domínio.

Na arquitetura de um sistema de gestão de projetos de software deve levar-se em consideração as suas restrições e a gestão administrativa. A administração de um sistema exige o conhecimento mais profundo do sistema, bem como quais as matrizes de rastreabilidade que o compõem. Neste sentido, foi analisado e implementado um

sistema de gestão administrativa que acumula todas as funcionalidades de um utilizador comum e as funcionalidades administrativas. A Figura 18 ilustra todas as funcionalidades de um Administrador do sistema.

Neste sistema de gestão e projetos de software foram implementadas as funcionalidades de gestão de utilizadores, gestão de habilidades e o caso excecional de alteração de *Project Manager* num projeto específico.

No caso da gestão de utilizadores o sistema foi modelado e implementado de forma a poder gerir vários utilizadores de sistema que poderão ou não possuir o perfil de administrador com as respetivas regras de rastreabilidade que lhe são imputadas na sua manutenção.

A gestão de habilidades (*Skills*) permite ao administrador gerir um conjunto de habilidades a importar por cada *Project Manager* e *Team Members* que compõem a equipa de projeto. Note-se que o *Project Manager* também é intrinsecamente um *Team Member* mas com a responsabilidade acrescida de líder de projeto.

De acordo com o PMBOK, a gestão de habilidades é distinguida conforme os intervenientes e a sua importância no desenvolvimento do sistema. Por isso os *Team Members* estão associados a habilidades específicas e os *Stakeholders* estão associados a papéis ocasionais e dependentes do âmbito do sistema.

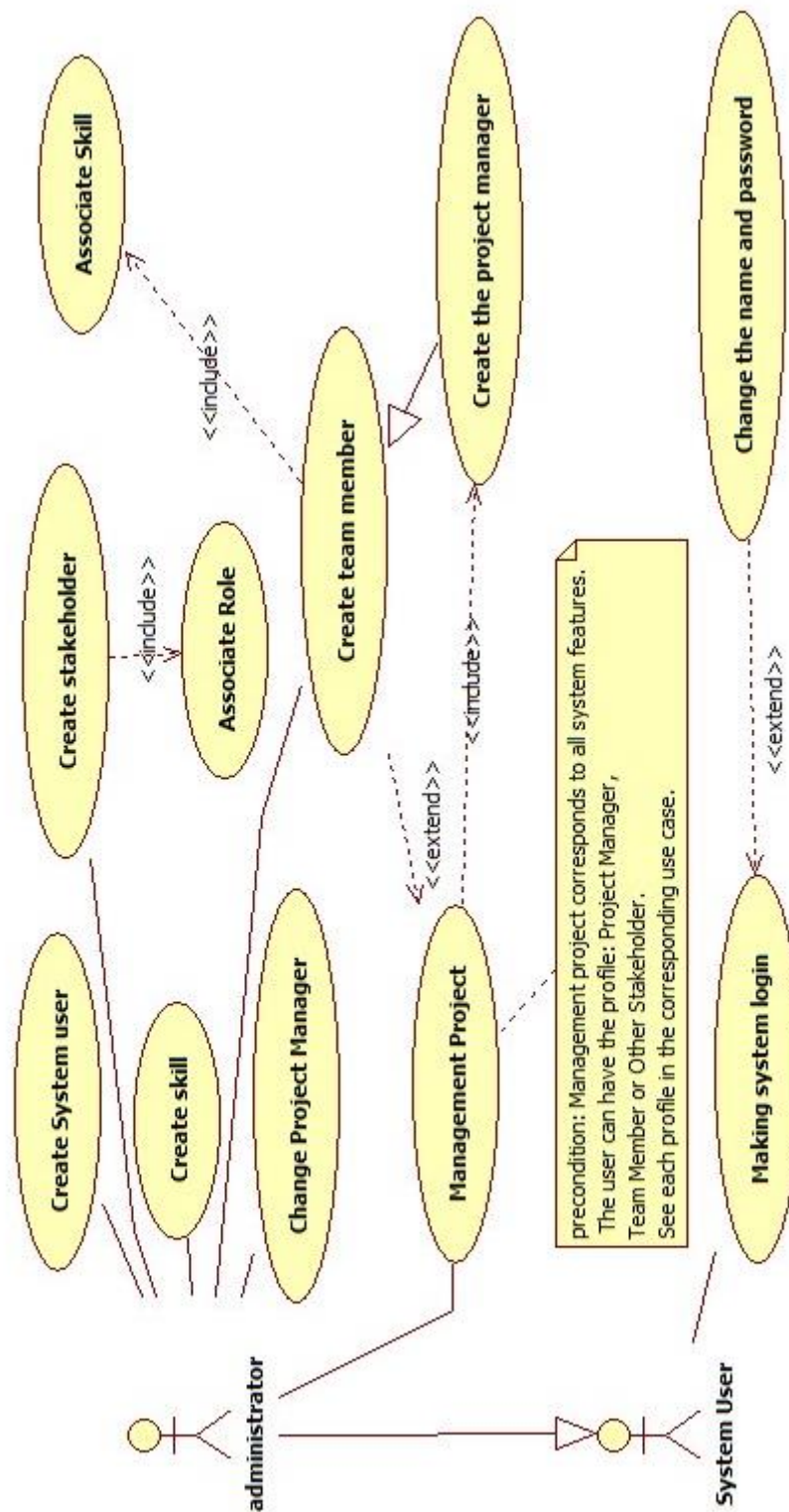


Figura 18 - Diagrama de casos de uso do Administrador do sistema.

Relativamente à mudança de *Project Manager* é uma funcionalidade de extrema responsabilidade e que permite garantir que o projeto tenha sempre um líder, mesmo que este desapareça no contexto do mundo real por circunstâncias maiores.

Neste contexto foi pensado, analisado e implementado um sistema que assegura a mudança de *Project Manager* em qualquer projeto, esteja ou não, o administrador, relacionado com o projeto em questão.

Todas as outras funcionalidades inerentes à gestão de projetos de software, estão implementadas e assumem o comportamento desejado, dependendo do papel interventivo, que o administrador tenha em cada projeto. Ou seja, o administrador, dentro de cada projeto, pode assumir um dos três papéis interventivos que já foram descritos anteriormente como sendo papéis da lógica de negócio.

Assim, independentemente das funções de administração, cada utilizador pode ser *Project Manager*, *Team Member* ou *Other Stakeholder*, dependendo do projeto em que está envolvido.

#### 4.5 MODELO DE DOMÍNIO

O diagrama de classes descreve os objetos e informações de estruturas usadas na aplicação, internamente e de comunicação com os seus utilizadores. Ele descreve as informações sem referência a qualquer implementação específica. As classes e relações servem de modelo para diferentes implementações, tais como tabelas em bases de dados, nós XML ou composições de objetos de software.

Tendo em vista a implementação, a gestão de dados e o fluxo de informação, foi modelado e definido o diagrama de classes (Figura 19) que permite a descrição total da

estrutura de objetos no sistema. Através de uma análise abstrata foram identificadas as classes de objetos (Entidades de Domínio) relevantes no contexto que se pretende modelar. Assim as classes descrevem objetos com atributos e operações comuns servindo dois propósitos. Por um lado permite compreender o mundo real naquilo que é relevante para o sistema que se pretende desenvolver e fornece uma base prática para sua implementação, quer seja a nível de programação por objetos, quer na definição da base de dados.

A Figura 19 apresenta o modelo de classes que dá suporte à aplicação modelada e desenvolvida para a gestão de projetos de software. De acordo com o diagrama e com as regras de negócio posteriormente implementadas no desenvolvimento, foi modelado o diagrama de classes definindo os respetivos relacionamentos entre as Entidades e a multiplicidade entre as classes definidas.

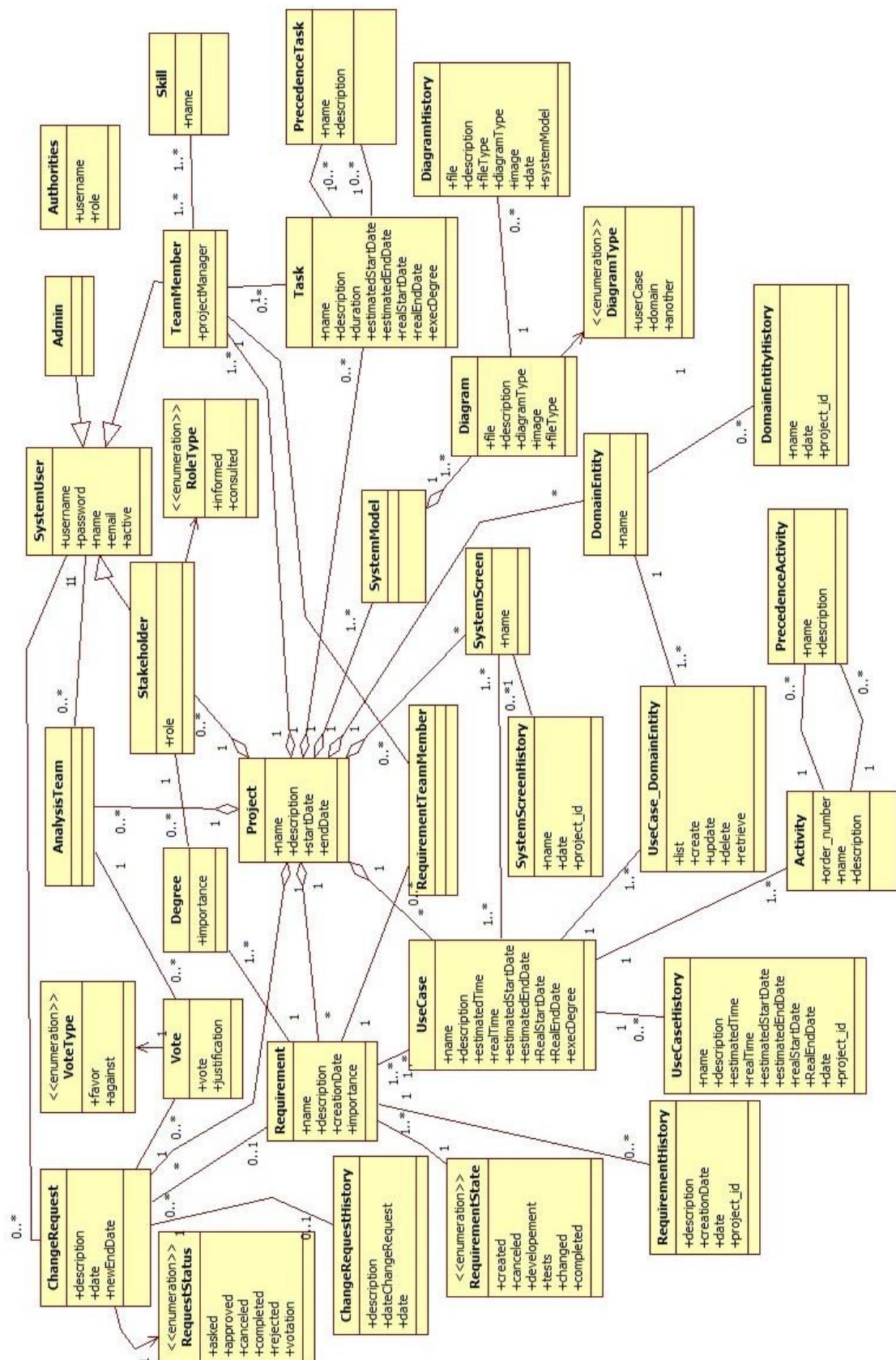


Figura 19 - Diagrama de Classes do Domínio do Projeto.

O modelo apresentado tem por base suportar a gestão de projetos de software com base no modelo PMBOK. Neste sentido, inclui outras Entidades para além das já definidas ou praticamente anunciadas pelas funcionalidades descritas nos *Use Cases* anteriormente apresentados.

A modelação do sistema inclui, para além de todas as entidades diretas do domínio, a gestão de históricos. A gestão de históricos é feita em: pedidos de alteração, requisitos, casos de uso, entidades de domínio, sistema de ecrãs e diagramas. Todo este conjunto de históricos permite ao sistema rastrear todas as alterações a requisitos, casos de uso, entidades e ecrãs de cada projeto.

No modelo PMBOK, é importante o valor que a informação tem em todas as suas etapas, mas a gestão ganha ainda mais eficiência e eficácia, quando se consegue otimizar processos ou relativizar situações de aparente complexidade. É nisto que o sistema de gestão de projetos, seja ele em que área for, se consegue destacar. E, tendo por base o modelo PMBOK, que pode ser aplicado na gestão de qualquer projeto, é possível acrescentar conhecimento se todas as áreas de conhecimento contribuírem uniformes para o objetivo comum, ou seja, o sucesso do projeto.

De acordo com o diagrama de classes apresentado, pode confirmar-se os tipos de utilizadores ao nível de administração. A tabela *“Authorities”* define através do username que tipo de utilizador terá o sistema aquando da sua autenticação. Foi definida isoladamente de acordo com a ferramenta *“Spring MVC”* de desenvolvimento, de forma a reaproveitar as potencialidades da framework ao nível da programação. Neste sentido sempre que um utilizador é criado na tabela *“System User”* é criada uma

entrada automaticamente na tabela *"Authorities"* o que facilita a gestão administrativa configurada com a segurança necessária.

Como apresentado anteriormente, o administrador é um simples utilizador com a responsabilidade de gerir todo o sistema.

De acordo com o modelo, a tabela *"SystemUser"* é a base para a criação de utilizadores e a gestão de interesses em cada projeto, por isso um *"Stakeholder"* e um *"Team Member"* são utilizadores distintos dentro do contexto do projeto. Como o *Stakeholder* representa alguém envolvido na equipa de projeto, é-lhe atribuído um *"Role"* que pode por exemplo, ser alguém apenas com a função de informar, não interagindo na decisão do projeto. Em contrapartida o *Team Member* é um utilizador que tem uma ou várias habilidades daí a relação com a tabela *"Skills"*.

A entidade *"Projet"* é a base de todo o sistema, depois da configuração de autenticação. Neste sentido, e dependendo de cada utilizador, são definidas operações e armazenados os dados necessários resultantes de operações durante o uso do sistema.

Tendo em conta a rastreabilidade das entidades, a entidade *"Project"* relaciona-se com as entidades *"Requirements"*, *"Task"*, *"UseCase"*, *"Stakeholder"*, *"Project Manager"*, *"Analysis Team"*, *"ChangeRequest"*, *"SystemScreen"*, *"DomainEntity"* e *"SystemModel"* muitas através de tabelas de relacionamento, de forma a permitir a multiplicidade das entidades envolvidas.

Atente-se à importância das tabelas *"Task"* e *"Activity"* que permitem, através das tabelas *"PrecedenceTask"* e *"PrecedenceActivity"* respetivamente, definir a precedência e a ordem com que as tarefas ou atividades são executadas. Na fase de



análise foi determinante perceber a sua funcionalidade de forma a permitir modelar o sistema.

Convém distinguir a importância dos conceitos de equipa de projeto e equipa de análise ao nível das classes. Neste sentido a tabela *“Team Member”* e a tabela *“Stakeholder”* constituem a equipa de projeto. A tabela *“AnalysisTeam”* permite criar a equipa de análise constituída pelos membros que fazem parte da equipa do projeto, ou seja, o conjunto de *Stakeholders* e *Team Members* do projeto.

A tabela *“AnalysisTeam”* assume outro papel preponderante, para além de criar a equipa de análise. Neste caso é através dela que surge a rastreabilidade para os pedidos de alteração ao projeto através da entidade *“ChangeRequest”*. Uma vez que o pedido de alteração pode sofrer alterações, e é aberto a toda a comunidade da equipa de análise, surge a entidade *“Vote”* onde cada interveniente da equipa de análise intervém de forma ativa nos pedidos de alteração aos requisitos ou âmbito do projeto.

A modelação levou em conta, como já foi referido anteriormente, a possibilidade de integrar diferentes diagramas, dependendo da importação do ficheiro relativo ao diagrama a ser implementado.

De acordo com os requisitos iniciais do sistema, foram modelados os históricos necessários à implementação do sistema de gestão de projetos. Neste sentido, foi analisada a possibilidade de alteração de requisitos, casos de uso, domínio de entidades, sistemas de ecrãs, diagramas e pedidos de alteração de forma a garantir a aprendizagem do sistema para projetos futuros, bem como monitorização das alterações efetuadas por cada utilizador em cada tarefa executada neste âmbito.

## 4.6 STORYBOARDS DO SISTEMA

Os Storyboards são modelos gráficos com uma série de ilustrações sequenciais que servem para dar corpo visual a uma implementação. É uma técnica bastante utilizada que permite a criação rápida de protótipos de baixo custo que, por sua vez, permitem a fácil validação com o cliente final. Com esta técnica o cliente pode visualizar e validar facilmente se os requisitos definidos, que dão origem àqueles ecrãs, vão ao encontro das necessidades que definiu.

Neste projeto foram desenvolvidos dois conjuntos de Storyboards, organizados por tipo de utilizador e por perfil administrativo, tendo em vista a implementação de uma aplicação que dê suporte à gestão de projetos de software.

São aqui apresentados apenas os Storyboards criados para a parte que foi devidamente implementada, e que dá corpo à aplicação web de gestão de projetos de software.

Assim como na estrutura dos casos de uso, os Storyboards da aplicação foram definidos por perfil de utilizador. Em anexos, pode ver-se com detalhe toda a sequência e o design esboçado, para a aplicação desenvolvida

Na sequência da análise realizada à modelação tendo em conta a finalidade da aplicação, foram definidos cada Storyboard composto por partes que podem ser desmontadas de forma a perceber o comportamento da aplicação.

O controlo de acessos é comum a todo o sistema, e permite através da entrada de username e password a autenticação do utilizador.

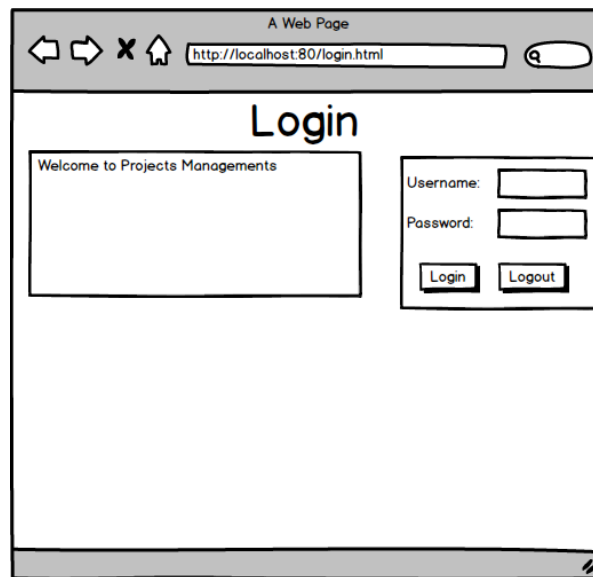


Figura 20 - Storyboard do login da aplicação.

O Storyboard “Login” (Figura 20), foi esboçado de forma a permitir que o utilizador se identifique no sistema, e este lhe dê as permissões necessárias para agir dentro dele.

Com a autenticação validada o utilizador passa a ter um de dois perfis ao nível administrativo, isto é, pode ou não ser administrador do sistema.

Assumindo que o utilizador é administrador, de acordo com o diagrama de casos de uso do administrador definido na Figura 18, Página 81, o sistema munir-se-á de três operações adicionais, categorizadas por “Users”, “Skills” e “Other Projects” na parte superior da barra de operações.

Ao nível gráfico a página apresentará um design semelhante ao apresentado na Figura 21, disponibilizando as operações administrativas desta aplicação. Com apenas um clique é possível aceder a qualquer uma destas operações.

A operação “Users”, permite aceder à lista de utilizadores que a aplicação possui (Anexos em Storyboard do Administrador na Figura 78). Através desta página é possível

fazer toda a manutenção ao nível do utilizador, podendo editar os campos relativos ao utilizador (Anexos em StoryBoard do Administrador na Figura 79).

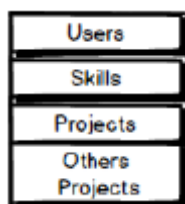


Figura 21 - Operações administrativas do projeto.

A operação “*Skills*”, permite aceder à lista de habilidades que a aplicação possui (capítulo Anexos em Storyboard do Administrador na Figura 80). Através da lista de habilidades é possível fazer a manutenção de todas as habilidades existentes ou criar novas habilidades de acordo com o projeto a definir (Anexos em Storyboard do Administrador na Figura 81).

A gestão de habilidades é muito importante. De acordo com o modelo PMBOK, todo o interveniente de um projeto deve ter o seu papel. É importante definir a este nível todas as habilidades requeridas pelo projeto para poderem ser atribuídas pelo gestor do projeto aos *Stakeholders* que irão fazer parte da equipa do projeto com o perfil *Team Member*.

A operação “*Other Projects*” permite aceder a todos os projetos com que o utilizador não está relacionado, ou seja, não faz parte da equipa de projeto desse projeto específico. Mas ao nível administrativo, é concedido ao administrador a possibilidade de poder alterar o gestor de projeto, pois pode, por questões excecionais, haver a necessidade forçada de proceder a essa alteração. Neste caso o administrador pode ver a informação básica do projeto e apenas alterar o gestor de projeto, definido na aplicação como “*Project Manager*” (Anexos em Storyboard do Administrador na Figura 84).

A operação “*Projects*” permite aceder facilmente a todos os projetos cujo utilizador faz parte da equipa de projeto, podendo ter o perfil de *Project Manager*, *Team Member* ou *Stakeholder* no projeto respetivo. Esta operação é comum a todos os casos de uso definidos, respetivamente para os perfis indicados na Figura 15, Figura 16 e Figura 17. Assim o diagrama de casos de uso do administrador herda as funcionalidades relativas à operação “*Projects*” bem como os casos de uso a ele associados. Observando todos os diagramas de casos de uso referidos anteriormente, incluindo o diagrama de casos de uso do administrador, verifica-se que o administrador é uma especificação de um utilizador, com esta operação em comum (Anexos, Storyboard do Administrador na Figura 82).

Quando o utilizador não tem o perfil de administrador do sistema, apenas é disponibilizado o acesso aos projetos a que ele está associado e a sua conta de utilizador, onde poderá configurar pessoalmente os seus dados. Para melhor compreender o processo ao nível do design esboçado e futura implementação para cada uma destas operações, atente Anexos em Storyboard do Administrador e visualizar toda a sequência deste Storyboard.

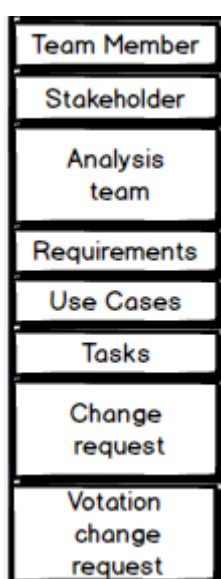


Figura 22 - Operações comuns do projeto.

Com o acesso aos projetos, a aplicação assume comportamentos diferentes para cada projeto, ao nível das operações a disponibilizar ao utilizador. Neste sentido, um utilizador que tenha o perfil *Project Manager* no projeto, tem ao seu dispor todo o conjunto de operações definidas de acordo com a Figura 22 anteriormente apresentada.

No entanto, o perfil do utilizador ao nível do projeto pode ainda variar, podendo ser *Team Member* ou ainda *Stakeholder*. Nestes casos, o leque das operações são mais reduzidas, de acordo com a lógica de negócio implementada. Para uma pequena apresentação gráfica, atente a Anexos em StoryBoard do *Project Manager*, StoryBoard do *Team Member* e StoryBoard do *Stakeholder* e seguir a sequência de processos de implementação do projeto.

No seguimento da construção do design, e baseados nos modelos de casos de uso anteriormente definidos, todos os utilizadores que não têm o perfil de administrador poderão aceder à gestão da sua conta de utilizador e à gestão dos projetos aos quais está afeto, através de um dos três perfis de utilizador de projeto (Anexos em StoryBoard do Project Manager na Figura 33).

O *Project Manager* tem à sua responsabilidade a criação da equipa de projeto, constituída por *Team Members* e *Stakeholders*. Neste caso através dos botões “*Team Member*” e “*Stakeholder*” poderá associar os respetivos utilizadores à equipa de projeto.

Segundo o modelo PMBOK, todos os *Stakeholders* devem ter um papel preponderante, sendo que, no caso dos especialistas, têm uma ou variadas habilidades de acordo com a atividade a desenvolver no projeto.

Definido no diagrama de casos de uso do Project Manager, o *Project Manager* pode associar e criar *Team Members* com uma ou várias habilidades (*Skills*).

O botão “*Team Member*” procede à operação que permite adicionar através de uma lista de utilizadores do sistema, os utilizadores que terão o perfil de *Team Member* no projeto (Anexos em StoryBoard do Project Manager na Figura 34). Cada *Team Member* pode ou não ser associado, à equipa de projeto de acordo com a escolha do *Project Manager*, sendo-lhe obrigatoriamente atribuído uma habilidade definida no dicionário de habilidades. (capitulo Anexos em StoryBoard do Project Manager na Figura 34).

Segundo o modelo PMBOK, os Stakeholders que não tenham uma componente técnica mas que tenham influência no desenvolvimento do projeto devem merecer interesse e serem valorizados no ceio do projeto. Neste caso são definidos como *Other Stakeholders* e tratados como “*Stakeholders Proponente*” no diagrama de casos de uso do *Stakeholder* proponente.

No referido diagrama de casos de uso, os utilizadores que são escolhidos pelo *Project Manager* para fazerem parte da equipa de projeto, é-lhe atribuído obrigatoriamente um papel (*Role*) que ele tem ao nível do projeto.

Com o botão “*Stakeholder*”, é possível listar os *Stakeholders* associados pelo *Project Manager* ao projeto (Anexos StoryBoard do Project Manager na Figura 36), bem como associar ou não cada *Stakeholder* e respetivo papel ao projeto (Anexos em StoryBoard do Project Manager na Figura 37).

De acordo com alguns guias de gestão de projetos, incluindo o modelo PMBOK, o grupo de *Team Members* e *Other Stakeholders* associados ao projeto, constituem a equipa de projeto. Todavia a equipa de projeto não é a equipa de análise, mas dela saem os membros que constituirão a equipa de análise. No modelo PMBOK a equipa de

análise assume um papel preponderante na sublimação de problemas de comunicação e tomadas de decisão complexas.

Neste sentido, o *Project Manager* é também responsável por criar a equipa de análise descrito no diagrama de casos de uso do *Project Manager*. Com o botão “*Analysis Team*”, o *Project Manager* cria a equipa de análise, que é obrigatoriamente criada com participantes da equipa de projeto. A operação “*Analysis Team*” permite listar e associar ou não os elementos da equipa de projeto à equipa de análise a construir (Anexos StoryBoard do Project Manager na Figura 38).

Constituída a equipa de análise do projeto, todo o utilizador afeto à equipa de análise pode criar requisitos (no caso de Project Manager ou *Team Member*) ou propor requisitos (no caso de *Other Stakeholder*). O modelo PMBOK define que cada requisito deve ser identificado por cada profissional ou proposto por cada elemento que faça parte da equipa de projeto. Nada deve ser deixado ao acaso, e toda a relação da informação pode tornar-se importante no cruzamento desta com uma comunicação explícita e sequenciada da lógica de negócio a implementar para atingir o objetivo do projeto.

No diagrama de casos de uso do Project Manager, pode observar-se o caso de uso referente à criação de requisitos por parte do utilizador. Assim todo o *Project Manager* pode definir os requisitos que achar necessários para a definição básica do projeto. Convínhamos que este caso de uso é partilhado também no diagrama de casos de uso do *Team Member* e transformado em sentido de proposição no diagrama de casos de uso do *Stakeholder*.

Ao nível do design, o *Project Manager* pode criar requisitos através de botão “*Requirements*”. Esta funcionalidade permite listar de imediato todos os requisitos



existentes no projeto em que se está a trabalhar (Anexos em StoryBoard do Project Manager na Figura 39), ou criar ou editar cada projeto de individualmente podendo neste último associar casos de uso que possam estar afetos ao respetivo requisito (Anexos em StoryBoard do Project Manager na Figura 40). É possível não estabelecer uma ordem de criação dos diferentes requisitos e cada funcionalidade a ele associado. No caso concreto, o utilizador no projeto pode criar em primeiro casos de uso seguido da criação de tarefas e requisitos ou estabelecer qualquer outra ordem, permitindo a liberdade total de definição e levantamento de requisitos ao nível da gestão do projeto. A mesma funcionalidade de edição de requisito permite sempre que necessário, associar o ou os casos de uso a ele associados, permitindo ao *Project Manager* ter uma visão global do relacionamento de cada requisito no âmbito das funcionalidades desenvolver, neste caso, monitorizar e controlar que casos de uso estão afetos ao requisito especificado.

De acordo com o modelo PMBOK, todos os requisitos são importante e a comunicação é fundamental para a uniformização de definição dos mesmos, não só para a definição da gestão de riscos mas também a calendarização e definição de timings dos projeto em desenvolvimento. Neste sentido é importante perceber que tipo de requisitos temos em mãos, que relação tem cada requisito com os casos de uso a ele associados, bem como monitorizar o estado de cada requisito. Portanto, é importante definir ou monitorizar os casos de uso que vão constituindo o projeto. Na ótica do *Project Manager* bem como na ótica do *Team Member*, os diagramas de casos de uso destes dois perfis apontam a necessidade de criação de casos de uso e a sua relação com os requisitos e suas tarefas associadas. Note-se que os utilizadores com perfil de

*Stakeholder* no respetivo diagrama de casos de uso do *Stakeholder* no projeto apenas está restrito a propor requisitos e associar casos de uso a ele relacionados.

A gestão de projetos de acordo com o modelo PMBOK, defende que à medida que o conhecimento é mais clarificado e portanto mais concreto, devem ser definidos ou refinados os casos de uso que modelam o mesmo, permitindo a este encontrar a forma real de funcionamento de implementação futura.

Neste sentido o *Project Manager* pode monitorizar todos os casos de uso afetos ao projeto em desenvolvimento no botão “*Use Cases*” (Anexos em StoryBoard do Project Manager na Figura 41) e gerir cada caso de uso, controlando e ou associando os requisitos e as tarefas afetas ao respetivo caso de uso (Anexos em StoryBoard do Project Manager na figura 42).

Ao nível da calendarização e timings é importante apontar datas marco para a criação de cada funcionalidade e que timing terá cada implementação, de forma a poder gerir o risco e o custo afeto a cada requisito e melhor orçamentar o projeto em desenvolvimento.

É importante recordar que, muitas vezes, os projetos falham por falta de gestão de timings ou deficiente orçamentação o que implica alargamento de prazos, falta de honra de compromissos inicialmente celebrados pelas partes (gestão de aquisições) ou mesmo má gestão de qualidade do produto em desenvolvimento.

De acordo com o modelo PMBOK, as tarefas a ser desenvolvidas por cada caso de uso devem merecer um interesse particular pois permitem detalhar cada função a concretizar no respetivo caso de uso e monitorizar e acompanhar o desenvolvimento de cada tarefa pelos respetivos recursos humanos. Desta forma as tarefas têm uma importante relevância ao nível do desenvolvimento podendo visualizar em que ponto se

encontra uma tarefa e que timing lhe resta para a sua concretização, evitando atrasos e derrapagens pontuais no projeto que poderão ser gravosas ao nível geral.

O *Project Manager* tem a seu cargo, distribuir tarefas pelos diversos *Team Members*, podendo os dois criar e definir tarefas e associá-las aos casos de uso que as envolvem. Atente-se ao diagrama de casos de uso do Project Manager e ao diagrama de casos de uso do Team Member. Os utilizadores com este perfil podem criar tarefas e associar-lhe os respetivos casos de uso.

Ao nível do design o *Project Manager* pode aceder à lista de tarefas a ele associadas através do botão “*Tasks*” (Anexos em StoryBoard do Project Manager na Figura 43). Cada tarefa pode ser criada pelo *Team Member* da equipa de projeto, o qual pode monitorizar o seu desenvolvimento e datas marco estimadas e reais (Anexos em StoryBoard do Project Manager na Figura 44). Neste sentido o *Project Manager* tem ao seu dispor uma forma fácil e apelativa de gerir as tarefas de cada requisito bem como toda a informação passível de análise e gestão tais como casos de uso e requisitos do respetivo projeto.

A gestão de projetos de software implica a gestão de diferentes recursos e a gestão de conflitos entre os diferentes recursos envolvidos, bem como a respetiva gestão dos diferentes tipos de *Stakeholders*.

No modelo PMBOK refere que a gestão de *Stakeholders* deve constituir um ponto fundamental na gestão de projetos bem como o seu alinhamento na gestão de recursos humanos que os envolvem. Neste sentido, o *Project Manager* deve saber gerir o conflito e criar formas de comunicação que favorece o desenvolvimento e avanço do projeto. Esta questão ganha ainda mais relevância no ceio da equipa de análise de um projeto onde reúne os intervenientes com poder de decisão e que conhecem de perto as

área específica do desenvolvimento. Assim, o *Project Manager* deve estar aberto ao diálogo e escutar os diferentes pontos de vista de cada membro da equipa de análise. Neste sentido e atendendo ao diagrama de casos de uso de *Project Manager*, diagrama de casos de uso do *Team Member* e diagrama de casos de uso do “*Stakeholder Proponente*”, cada membro da equipa de análise, pode solicitar alterações de requisitos no projeto, podendo este ser aceite ou não pela equipa de projeto.

Ao nível do design, o botão “*Change Request*” permite ao *Project Manager* monitorizar todos os pedidos solicitados e alterar o seu estado de acordo com a decisão tomada pela equipa de análise. Neste sentido o *Project Manager* tem a seu cargo a responsabilidade de alterar o estado de cada pedido de alteração (Anexos em StoryBoard do Project Manager na Figura 48), podendo este, estar no estado ASKED, APPROVED, CANCELED, COMPLETED, REJECTED,VOTATION, como se pode visualizar no diagrama de classes no modelo de domínio.

Para criar um pedido de alteração ao requisito por parte de cada membro da equipa de análise basta apenas definir o requisito que pretende alterar ou potencialmente uma data marco e solicitar o mesmo pedido à respetiva equipa que analisará que pronunciar-se-á favorável ou contra a mesma solicitação, sendo que o *Project Manager* tem sempre a última palavra e portanto a decisão final (Anexos em StoryBoard do Project Manager na Figura 49).

No seguimento da gestão da equipa de análise e como já foi referido o *Project Manager* tem um papel preponderante na gestão de toda a equipa de decisão, mas, segundo o modelo PMBOK, deve dar-se a atenção a cada membro da equipa de análise e escutar todos os seus pontos de vista e suas respetivas justificações. Neste sentido e atendendo à cordialidade democrática o diagrama de casos de uso do *Project Manager*,

bem como dos outros dois tipos de perfis que têm sido explicados, apresentam o caso de uso referente à aprovação de mudanças de requisitos.

Em termos aplicacionais, o design foi definido de forma uniforme permitindo ao *Project Manager* através do botão “*Votation*” aceder à lista de pedidos de alteração de requisitos que estão à espera de serem analisados e sujeitos à decisão de cada elemento constituinte da equipa de análise (Anexos em StoryBoard do Project Manager na Figura 50). O *Project Manager*, assim como cada elemento da equipa de análise, manifesta a sua opinião e justifica-a de forma a gerir a aprendizagem ao nível do projeto e contrapor os diferentes pontos de vista que certamente surgiram por parte de cada membro constituinte (Anexos em StoryBoard do Project Manager na Figura 51). No caso do *Project Manager* especificamente ao nível aplicacional, pode monitorizar cada opinião de cada membro da equipa de análise sendo estes restritos à visualização de cada tomada de decisão que são chamados. Desta forma, o projeto garante a conformidade da uniformização da comunicação entre os diferentes membros da equipa de análise e favorece o crescimento ao nível da aprendizagem pessoal, profissional e da equipa de cada requisito específico do projeto em desenvolvimento.

Na ótica do *Team Member*, tendo em conta as suas características de perfil, tal como foi referenciado anteriormente no perfil de *Project Manager*, o *Team Member* pode efetuar todo o conjunto de operações que o *Project Manager*, exceto constituir equipas, quer estas sejam de projeto ou de análise. Atente ao diagrama de casos de uso do *Team Member* e verifica-se que o *Team Member* pode, criar requisitos, criar casos de uso, criar tarefas e propor e aprovar alterações a requisitos se este fizer parte da equipa de análise.

Neste sentido e de acordo com o modelo PMBOK, o *Team Member* é um membro constituinte da equipa de projeto que pode ou não fazer parte da equipa de análise e que tem conhecimentos específicos em áreas de conhecimento e portanto relevância ao nível do desenvolvimento.

No seguimento do design definido o *Team Member* através do botão “*Requirements*” acede a todos os requisitos associados ao projeto (Anexos em StoryBoard do Team Member na figura 55), podendo fazer a gestão de cada requisito (Anexos em StoryBoard do Team Member na Figura 56) a ele associado.

Os casos de uso podem ser acedidos através do botão “*Use Cases*” que permite listar todos os casos de uso do projeto a que o *Team Member* está associado (Anexos em StoryBoard do Team Member 57). Cada caso de uso pode ser editado ou criado pelo *Team Member* envolvido, permitindo associar a ele todos os requisitos e tarefas relacionadas (Anexos em StoryBoard do Team Member na Figura 58).

No caso das tarefas, o *Team Member* envolvido pode aceder às tarefas a ele atribuídas através do botão “*Tasks*” (Anexos em StoryBoard do Team Member na Figura 59). Cada tarefa pode ser gerida de acordo com o *Team Member*, podendo este acrescentar e planificar novas tarefas de acordo com os casos de uso definidos e as necessidades do cliente (Anexos em StoryBoard do Team Member na Figura 60). É importante perceber que todas as tarefas definidas são monitorizadas e permitem perceber em termos de timing e grau de desenvolvimento em que situação se encontra a tarefa a ser desenvolvida.

A gestão de requisitos, nos *Team Members* envolvidos é muito importante, pelo peso que cada interveniente pode ter na sua gestão. Neste sentido os *Team Members* que são selecionados pelo *Project Manager* para fazer parte constituinte da equipa de

análise, são chamadas a outras responsabilidades já descritas anteriormente. Um *Team Member* que faça parte da equipa de análise pode propor alterações a requisitos à medida que o projeto de levantamento de requisitos avança, bem como intervir nas decisões de mudança dos mesmos.

Relativamente aos pedidos de alteração de requisitos, o *Team Member* pode aceder a todos os pedidos de alteração de requisitos através do botão “*Change Request*” que permite monitorizar a lista de pedidos de alteração do requisitos bem como o estado que têm no respetivo momento (Anexos em StoryBoard do Team Member na Figura 64). Note-se que na ótica do *Team Member*, não é possível alterar os estados de cada pedido de alteração ao requisito, esta funcionalidade é restrita ao *Project Manager*. Cada pedido de alteração do requisito proposto pelo *Team Member* pode ser monitorizado individualmente e gerido caso o seu estado seja o inicial, ou seja, o estado “ASKED” (Anexos em StoryBoard do Team Member na Figura 65).

Para que o *Team Member* possa intervir nos vários pedidos de alteração ao qual é chamado a intervir deve aceder à lista de pedidos de alteração no estado pedido através do botão “*Votation*” (Anexos em StoryBoard do Team Member na Figura 66). Neste nível o *Team Member* tem ao seu dispor todos os pedidos de alteração que estão pendentes da sua tomada de decisão. Assim, o *Team Member* pode aceder ao pedido de alteração no estado de votação pretendido e dar a sua opinião ou ponto de vista de forma favorável ou não, justificando a sua decisão (Anexos em StoryBoard do Team Member na Figura 66).

Desta forma o *Team Member* pode intervir e monitorizar todas as suas intervenções no ceio do projeto.

Relativamente ao perfil de *Stakeholder*, refere-se a todos os intervenientes no projeto que normalmente não têm conhecimentos específicos académicos na área do desenvolvimento do projeto, neste caso concreto ao nível da análise e desenvolvimento de software mas que de forma direta ou indireta contribuem para a sua modelação. Muitas vezes estes intervenientes, possuem informações determinantes para a definição de base do projeto. A gestão da comunicação mais uma vez é o ponto fundamental para extrair o que é importante para o desenvolvimento do projeto. Neste sentido o modelo PMBOK, considera a necessidade de dar distinta relevância aos diferentes tipos de *Stakeholders*. No caso específico do projeto a desenvolver atente ao diagrama de casos de uso do *Stakeholder*. Nele são definidas as funcionalidades idênticas às dos outros perfis anteriormente descritos mas com mais restrições.

Neste sentido o “*Stakeholder Proponente*” como é definido apenas pode propor requisitos, associa-los a casos de uso quando estes têm conhecimentos clarificados do domínio do projeto. O *Stakeholder* pode também solicitar alterações aos requisitos definidos e intervir na sua decisão quando este fizer parte da equipa de análise.

O *Stakeholder* pode por isso, aceder aos requisitos do projeto através do botão “*Requirements*” e aceder à lista de todos os requisitos do projeto (Anexos em StoryBoard do Stakeholder na Figura 71).

Para propor ou visualizar os requisitos criados, pode aceder a cada requisito individualmente através da lista de requisitos do projeto (Anexos em StoryBoard do Stakeholder na Figura72). Desta forma o sistema garante a comunhão da informação por parte de todos os intervenientes e a sua definição.

Quando o *Stakeholder* pertence à equipa de análise, este pode propor a alteração de requisitos, através do botão “*Change Request*”. Assim o *Stakeholder* pode



aceder a todos os pedidos de alteração dos requisitos propostos (Anexos em StoryBoard do Stakeholder na Figura 73) pelos diferentes intervenientes da equipa de análise. Cada pedido de alteração apresenta o seu estado, podendo este ser gerido por parte do *Stakeholder* se tiver sido criado por ele e ainda se encontrar no estado “ASKED” (Anexos em StoryBoard do Stakeholder na Figura 74).

No seguimento dos outros perfis pertencentes à equipa de análise, o *Stakeholder* pode intervir na tomada de decisão de pedidos de alteração de requisitos. Para aceder a todos os requisitos que estão no estado de votação a aplicação dispõe o botão “*Votation*” onde são apresentados todos os pedidos de alteração e o estado dos mesmos em que o *Stakeholder* participou.

Cada pedido de alteração pode ser acedido pelo *Stakeholder* que o criou de forma a poder editá-lo caso este seja criado pelo mesmo. Ou acedido para intervir manifestando o seu ponto de vista de forma justificada (Anexos em StoryBoard do Stakeholder na Figura 76).

Na linha dos outros perfis, o *Stakeholder* pode sempre manifestar a sua opinião e monitorizar o estado do pedido de alteração. No entanto assim como o *Team Member* a monitorização ao nível destes perfis é restritiva em comparação com o *Project Manager* que devem monitorizar e controlar todos os pontos de vistas e as justificações que levaram a essa tomada de decisão de formalizar a tomada de decisão final.

#### 4.7 TAREFAS AUTOMÁTICAS DO SISTEMA

O projeto apresentado, sempre baseado no modelo ou guia de boas práticas PMBOK, deve permitir a gestão de históricos nos diferentes domínios específicos de

forma a garantir a comodidade das diferentes áreas de conhecimentos ao não só no presente mas também no futuro. Ou seja, todos os projetos em desenvolvimento devem permitir uma aprendizagem a nível de gestão, neste sentido, não são só os recursos humanos que aprendem com a experiência mas toda a estrutura de conhecimento.

Com efeito a aplicação apresentada foi analisada e modelada de forma a garantir toda esta configuração. Atente-se ao diagrama de casos de uso de Project Manager, onde pode identificar-se todas as operações a implementar. Atente-se ao diagrama de classes onde se pode perceber a importância da gestão de históricos.

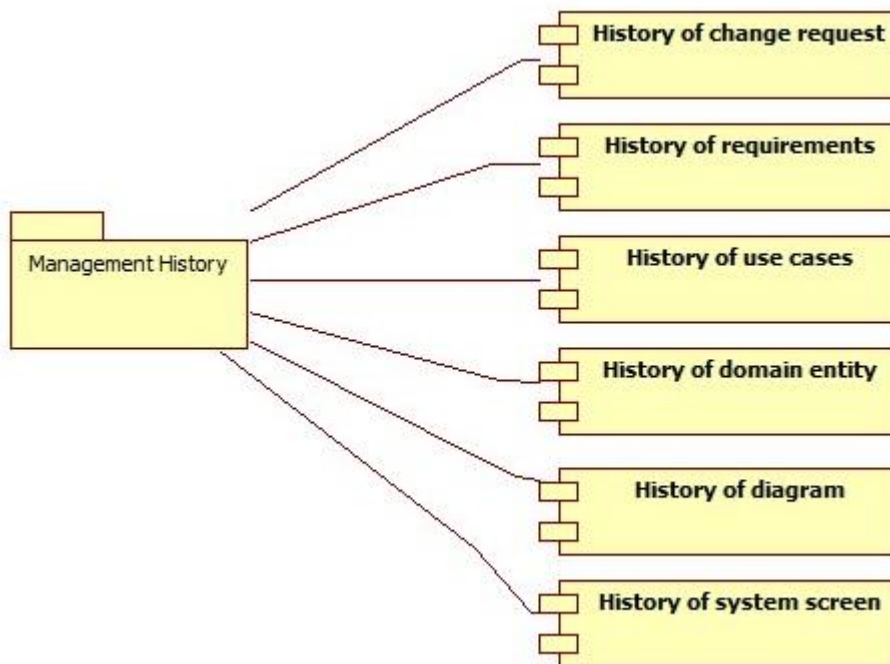


Figura 23 - Composição do pacote de gestão de históricos.

O histórico configura todas as operações realizadas pelo sistema automaticamente conforme o desenrolar da aplicação. Nele recai a responsabilidade de gerir todos os históricos associados a cada processo do sistema. Neste sentido e de acordo com a Figura 23 imediatamente acima apresentada, a gestão de históricos divide-se em seis subcomponentes que são: histórico de pedido de alteração, histórico

de requisitos, histórico de casos de uso, histórico de entidades de domínio, histórico de diagramas e históricos de sistemas de ecrãs.

O histórico de pedidos de alteração é inicializado sempre que um pedido de alteração é criado, guarda todas as alterações feitas ao pedido de alteração. É importante perceber em futuros projetos ou na identificação de informação no projeto em desenvolvimento em que condições ocorreu a mudança de estado e quem os criou.

O histórico de requisitos permite iniciar-se com a criação do requisito por parte de qualquer elemento da equipa de projeto e mantém-se até à conclusão do projeto, armazenando todos os estados do requisito e identificando quem procedeu à sua alteração.

O histórico de casos de uso é outro histórico que inicia-se com a criação do caso de uso do projeto e mantém-se até à conclusão do projeto, armazenando todas as alterações que o caso de uso vai tendo ao longo do seu ciclo de vida, nomeadamente monitorizando os requisitos e as tarefas e ele atribuídas.

O histórico de entidades de domínio inicia-se com a criação da entidade de domínio por parte dos intervenientes do projeto e mantém toda a informação ao longo de todo o projeto até à sua conclusão. Permite identificar as entidade de domínio inicialmente identificadas e qual a sua evolução ao longo do desenvolvimento do projeto.

O histórico de diagramas inicia-se com a criação do primeiro diagrama por parte dos intervenientes da equipa de análise e permite armazenar que evolução o sistema sofreu ao longo do projeto, ao nível da sua definição. Convém lembrar que os diagramas podem ser importados de outras ferramentas como o starUML pelo que a sua gestão é feita através da alteração de várias versões.

Por fim, o histórico de sistemas de ecrãs, que inicia-se com a criação do primeiro ecrã para o sistema em desenvolvimento e guarda todas as alterações até à versão de conclusão do projeto.

A gestão de históricos é uma parte que não foi desenvolvida. No entanto convém ressaltar que o sistema está modelado de forma a permitir a gestão do histórico e garantir a comodidade de todos os históricos em desenvolvimento. Parte do sistema ganha importância com o histórico em implementação, pois a gestão da aprendizagem, a gestão do risco e da qualidade é conseguida recorrendo ao historial das diferentes etapas que os diferentes projetos foram sofrendo. Neste sentido a gestão de histórico tem um valor preponderante a longo prazo ao nível da gestão de projetos de software.

#### 4.8 CONCLUSÕES OU NOTAS FINAIS

A gestão de projetos de software é uma área de grande complexidade e não possui um modelo exato de sucesso porque depende de vários fatores, nomeadamente os recursos humanos, e em específico o líder que dirige todo o processo. No entanto, há vários modelos que a serem seguidos garantem a sincronização e o mais provável sucesso do projeto.

Um líder destaca-se pela sua experiência e características, como a competência técnica, humana e deontológica. A sua responsabilidade comunicativa é fundamental para o sucesso do projeto.

Este projeto foi modelado e desenvolvido tendo por base o modelo PMBOK, que permite definir de uma forma técnica e metodológica todos os processos relativos à gestão de projetos.

Foram definidos e modelados todos os requisitos para uma aplicação web que dê suporte à gestão de projetos, com base na criação de diagramas de casos de uso, criação de diagramas de pacotes e respectivos Storyboards com vista à sua implementação aplicacional.

Neste sentido e com base no modelo PMBOK, conclui-se que num projeto de software é importante a gestão de *Stakeholders* e estes dividem-se em três tipos. Os tipos de *Stakeholders* são: o *Project Manager*, *Team Member* e *Other Stakeholders*.

Em termos de gestão, o conjunto dos *Stakeholders* é identificado como sendo composto por intervenientes do projeto. Assim, cada interveniente pode assumir um dos três tipos de perfis de *Stakeholders* no projeto, com diferentes graus de rastreabilidade.

O *Project Manager* é considerado um elemento da equipa de projeto mas com responsabilidades maiores, tais como a criação da equipa de projeto e da equipa de análise, para além de monitorizar e distribuir tarefas entre os diferentes intervenientes bem como escutar e decidir aspetos relevantes em relação aos requisitos a serem definidos no projeto.

O *Team Member*, são membros da equipa de projeto com conhecimentos técnicos especializados que apoiam o Gestor de projeto nas diferentes decisões e tarefas ao longo do projeto.

O *Other Stakeholders* são intervenientes da equipa de projeto que podem não ter conhecimentos técnicos especializados mas são fundamentais no desenrolar do projeto. Neste sentido são intervenientes que têm um papel cooperativo com a equipa de projeto ao longo do seu desenvolvimento.

A equipa de projeto é composta por todos os intervenientes do projeto e a equipa de análise é composta por alguns dos intervenientes da equipa de projeto com capacidade de decisão.

Ao nível das tarefas, todos os intervenientes podem criar requisitos, ao longo do projeto, e apenas os intervenientes com conhecimentos técnicos podem e devem contribuir na criação de tarefas e casos de uso bem como a sua calendarização.

A um nível mais especificado os intervenientes de projeto com responsabilidade ao nível da tomada de decisão, podem intervir e influenciar as alterações propostas por todos os intervenientes da equipa de análise.

Neste capítulo foram ainda analisadas e modeladas as potencialidades ao nível da gestão de históricos e criação de entidades de domínio, sistema de ecrãs e diagramas no sentido de munir o sistema de escalável capacidade de aprendizagem ao longo do tempo. Depois de apresentados todos os requisitos definidos e modelados, foram distribuídos ao longo do tempo desde a sua iniciação até a sua real implementação.

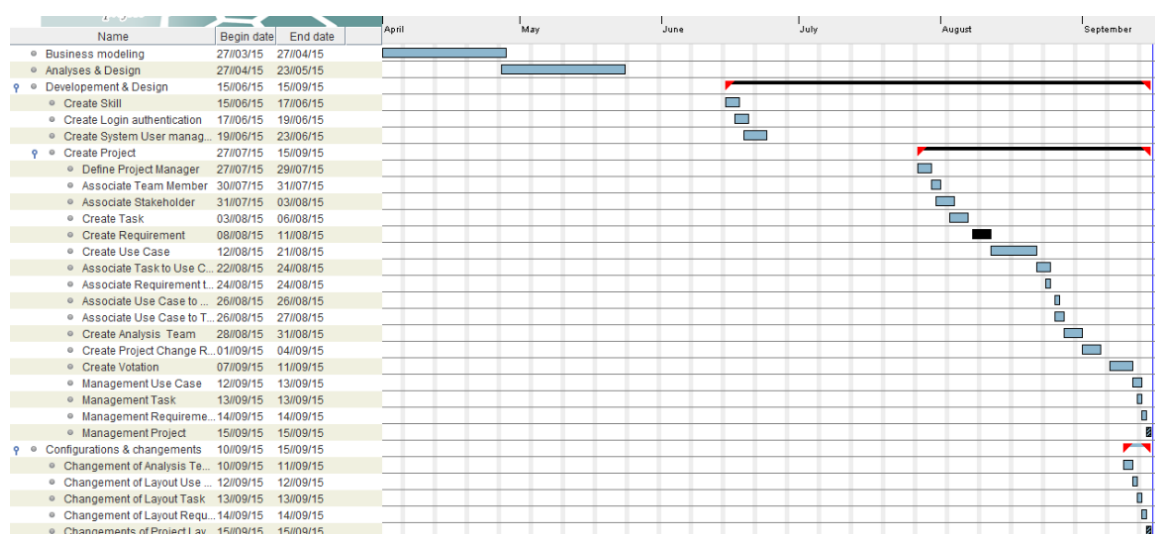
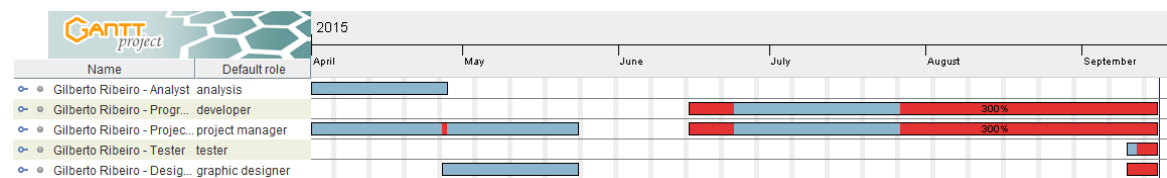


Figura 24 - Cronograma do Projeto.

De acordo com as diferentes etapas, desde o levantamento de requisitos à sua implementação, foi elaborado um cronograma por tarefas do desenvolvedor, apresentado na Figura 25.



**Figura 25 - Cronograma de tarefas por utilizador do projeto.**





## 5. DESENVOLVIMENTO DO PROJETO

### 5.1 INTRODUÇÃO

A importância da gestão de projetos tem tido grandes desenvolvimentos devido à necessidade cada vez maior de congregar o conhecimento com os intervenientes que o detêm.

A necessidade de alinhar uma ferramenta gráfica com um modelo de gestão de projetos é uma realidade e essencial para obter os resultados compatíveis com os objetivos a que um projeto de gestão se submete.

Nesta secção é apresentada uma aplicação gráfica com recurso a ferramentas Open Source. O exemplo consiste numa aplicação para gestão de projetos de software baseado nos princípios do guia/modelo PMBOK. Dada a complexidade do modelo e a abrangência analisada e modelada nos capítulos anteriores, são apresentados os extratos da aplicação, na sequência dos Storyboards também apresentados no capítulo anterior e ilustrados em Anexo A.1 e A.2 nesta tese de mestrado.

A aplicação foi desenvolvida em Java Spring MVC com postgresSQL e tem como principal objetivo ilustrar e dar suporte á gestão de projetos de software de uma forma apelativa, através de uma ferramenta gráfica web.

### 5.2 DESENHO DA APLICAÇÃO

Relativamente ao desenho da aplicação, esta é composta por nove partes essenciais, as quais se designam por (Figura 26):

- **Application Header** – O cabeçalho da aplicação, normalmente define uma barra de apresentação.

- **Logged user** – O user que está a operar.
- **Information operation running** – Informação relativa às operações que estão a ser executadas no momento. Lista a tarefa ou requisito a ser editado por exemplo.
- **Administrative informations** – Informação relativa às operações a serem executados pelo administrador.
- **Buttons' area for administrative operation** – Nesta área encontram-se os botões que permitem a gestão de operações administrativas. Neste caso encontram-se os botões relativos à gestão de utilizadores, gestão de habilidades e lista de projetos com que o *User* (administrador de sistema) autenticado não tem qualquer relação.
- **Operational information** – Informação relativa às operações a executar pelo *User* comum, ou seja, pelo *Project Manager*, *Team Member* ou *Stakeholder*.
- **Buttons' area for user operation** – Nesta área encontram-se os botões que permitem ao utilizador dependente do perfil no projeto em desenvolvimento, realizar as operações a ele atribuídas.
- **Current view** – Lista do conteúdo da operação em execução. Permite a interação entre o utilizador e a aplicação através de entrada e saída de dados.
- **Application footer** – Rodapé da página, onde é definido o copyright da página.

Dentro do contexto do projeto, a aplicação foi desenhada de forma a comportar uma área de botões de apoio aos perfis de cada utilizador e uma área de conteúdos onde é feita a interação entre o utilizador e o sistema.

Toda a área de conteúdos é uniforme de acordo com o perfil do utilizador, transmitindo o feedback necessário ao mesmo que está a trabalhar. Neste sentido esta área ajusta-se dinamicamente ao perfil de utilizador que está a trabalhar, facultando uma fácil utilização.

Tendo em consideração esta estrutura, a aplicação pode ser decomposta nas áreas apresentadas na Figura 26.

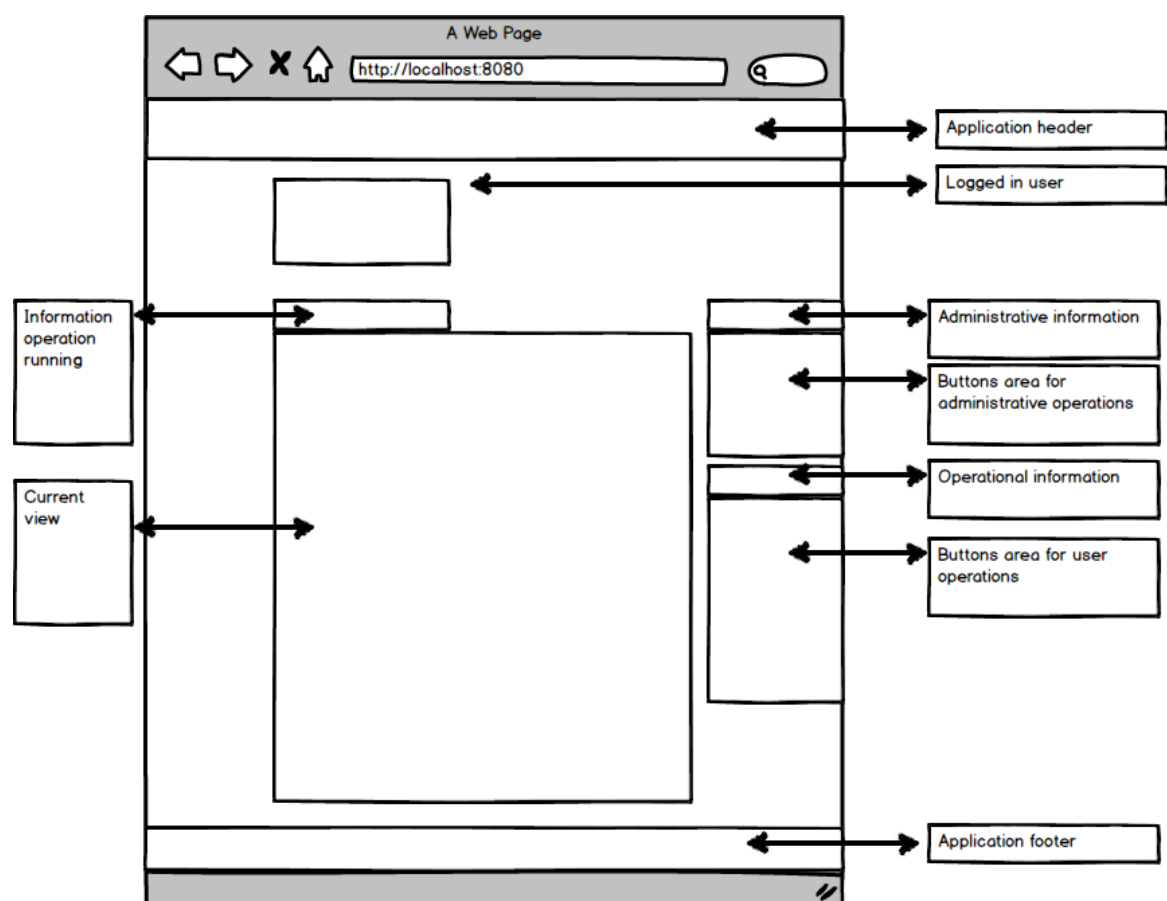


Figura 26 - Decomposição estrutural de aplicação.

### 5.3 COMPORTAMENTO

O comportamento da aplicação é determinado pelos perfis que a aplicação comporta no desenvolvimento de cada projeto e as operações a eles associados.

A aplicação desenvolvida pode ser dividida em quatro grandes perfis de utilização:

- **Administrador** – A área específica para administrar do sistema. Esta área apresenta botões específicos à gestão de todo o projeto, tais como: lista de habilidade (*Skills*), lista de utilizadores (*Users*), lista de outros projetos (*Other Projects*), onde permite alterar o gestor de projeto quando necessário.
- **Project Manager** – A área específica ao gestor do projeto, que apresenta botões com operações a desenvolver por um gestor de projeto (*Project Manager*).
- **Team Member** – Comporta botões com operações restritas de um membro de uma equipa de projeto.
- **Stakeholders** – A área de apoio aos intervenientes que podem apoiar o desenvolvimento do projeto mas não na estrutura técnica.

No seguimento de cada perfil abordado anteriormente, a aplicação ajusta a sua apresentação gráfica e o seu comportamento, pois de acordo com o modelo PMBOK, todos os intervenientes devem estar em sintonia ao nível da equipa de trabalho. Com uma abordagem uniforme a integração e adaptação de cada conceito tais como equipa de projeto, equipa de análise, relevância de diferentes intervenientes, definição de habilidades são enquadrados nos diferentes perfis, respeitando a mesma estrutura aplicacional.

Authentication information:

Enter your authentication credentials. Please login and manage your projects.

If you do not have authentication credentials. please request them from the manager or service provider!

Authentication

User Name :  
miguelAntunes

Password:  
\*\*\*\*\*

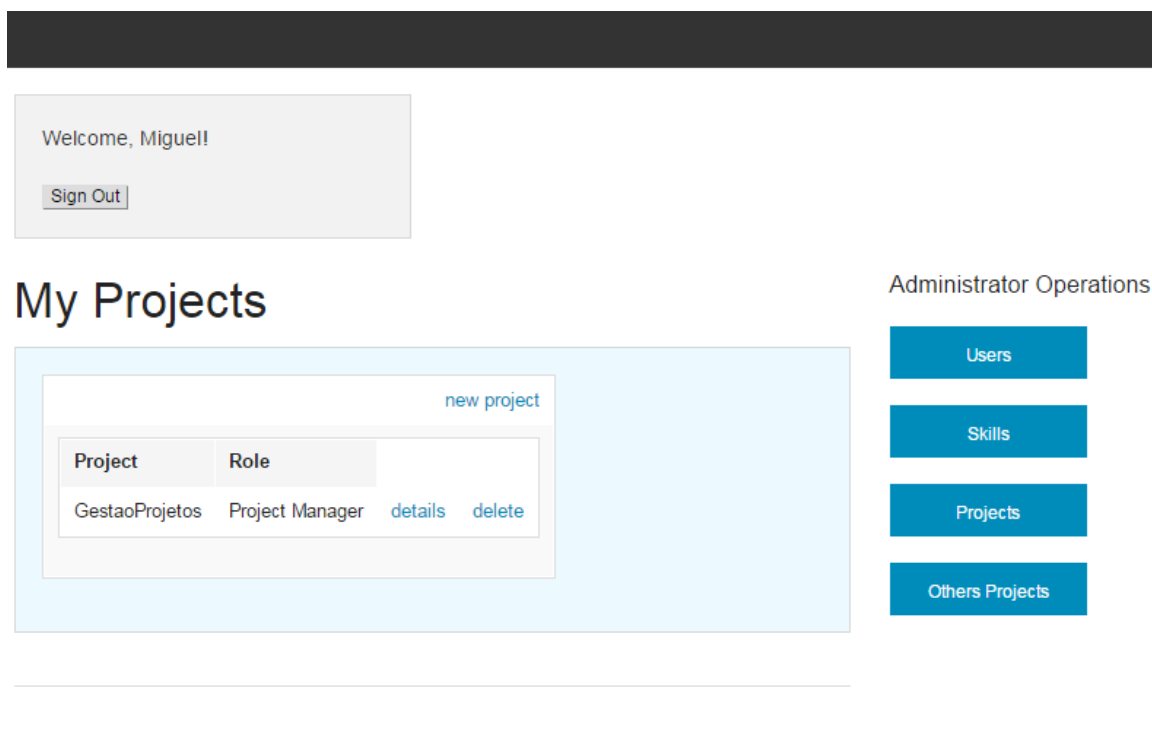
Sign In

© Copyright2015 Admin Footer Page

**Figura 27 - Administrator - Login da aplicação.**

A Figura 27 apresenta o login da aplicação, onde o utilizador é identificado se pertence ou não ao sistema e qual o seu papel dentro da aplicação, isto é, se tem ou não permissões administrativas.

Considerando que estamos na presença de um utilizador com permissões de administrador, então a aplicação disponibiliza de imediato as funções administrativas.



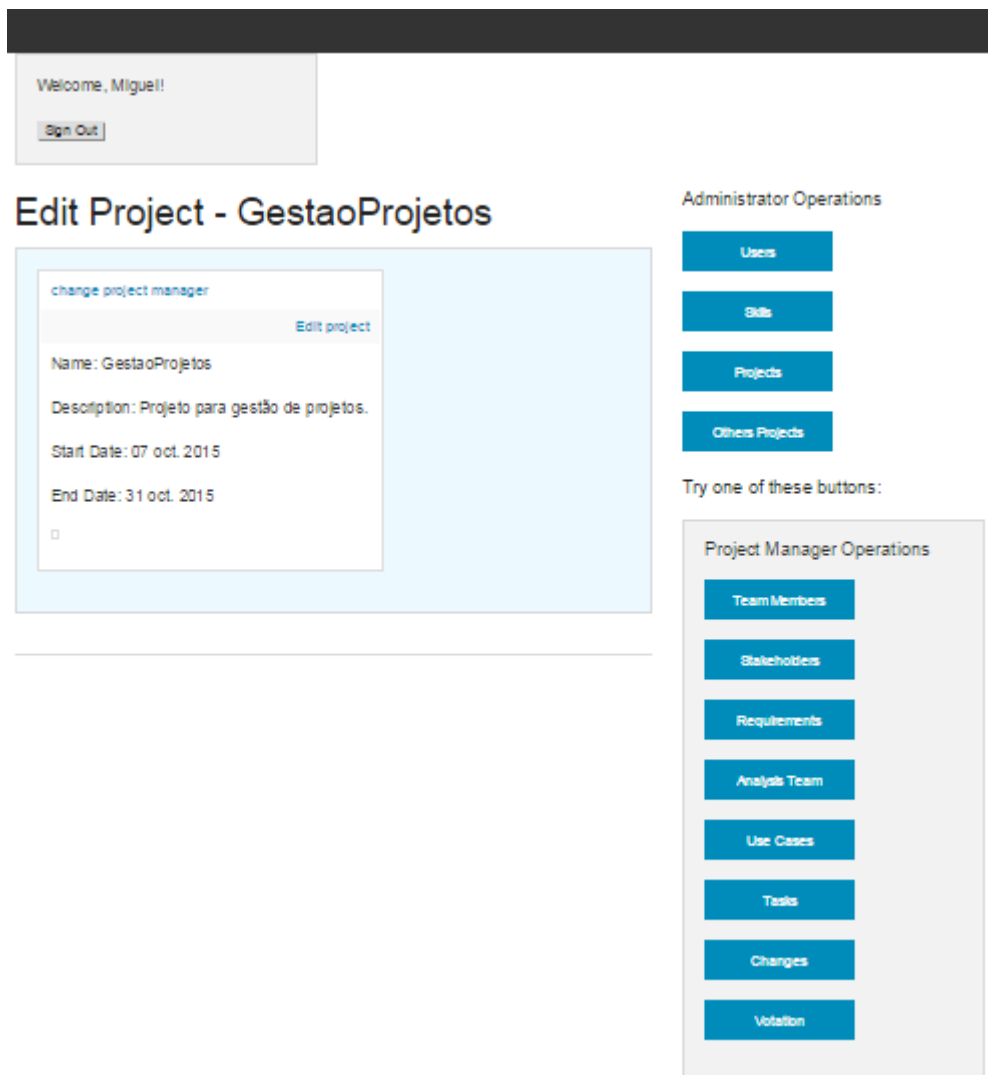
**Figura 28 - Administrator - Operações Administrativas.**

De acordo com a Figura 18, o administrador, suportado pela estrutura definida no desenho da aplicação, tem ao seu dispor através de botões a sua área de gestão conferida pela aplicação, resultado da sua autenticação. Neste sentido pode gerir: os utilizadores, as habilidades e as mudanças do gestor de projeto do sistema, respetivamente. Os detalhes de cada uma das operações podem ser observados na secção B em Screens do Administrador.

É importante relembrar que o modelo PMBOK releva a importância do papel que cada um dos intervenientes pode ter no projeto, por isso, todos os intervenientes devem merecer a atenção em diferentes domínios independentemente do perfil que tenham no ceio do projeto. Neste sentido considera-se que qualquer utilizador pode ser responsabilizado através da sua envolvência no projeto assumindo um perfil de responsabilidade de colaboração na equipa do projeto.

Atendendo a que a equipa de projeto é constituída por todos os intervenientes que colaboram diretamente no projeto, estes têm o seu perfil atribuído ao projeto em que colaboram, por isso, as funções administrativas devem ser independentes das funções de gestão de projetos.

A Figura 29, demonstra a independência entre as funções administrativas e as funções de gestão de projetos, bem como a dependência dos diferentes perfis do utilizador no projeto. No caso prático o administrador, assume um perfil no projeto quando afeto a este. Neste caso, foi responsável por criar um novo projeto, o qual assumiu o papel de *Project Manager*. A aplicação disponibiliza portanto operações específicas ao *Project Manager*, tais como a criação da equipa de projeto associando membros de equipa e *Stakeholders*, bem como a criação da equipa de análise para apoio nas tomadas de decisão. Além da Figura 29 apresentada, pode observar-se estas operações específicas na secção Anexos B em Screens do Project Manager.



**Figura 29 - Administrator - Operações do Project Manager.**

De acordo com o modelo PMBOK, os intervenientes do projeto podem ser vários e assumir cada um, um papel preponderante no desenrolar do projeto para atingir o seu objetivo, que é o sucesso do projeto.

Com a importância de manter a coordenação da comunicação entre os diferentes intervenientes, a monitorização das responsabilidades é um meio usado como recurso na evolução dos sistemas de gestão de projetos.

Um membro da equipa de projeto tem portanto responsabilidades diferentes, que um gestor de projeto. Neste sentido, deve contribuir com os seus conhecimentos, em cada etapa de desenvolvimento do projeto a ele destinada.



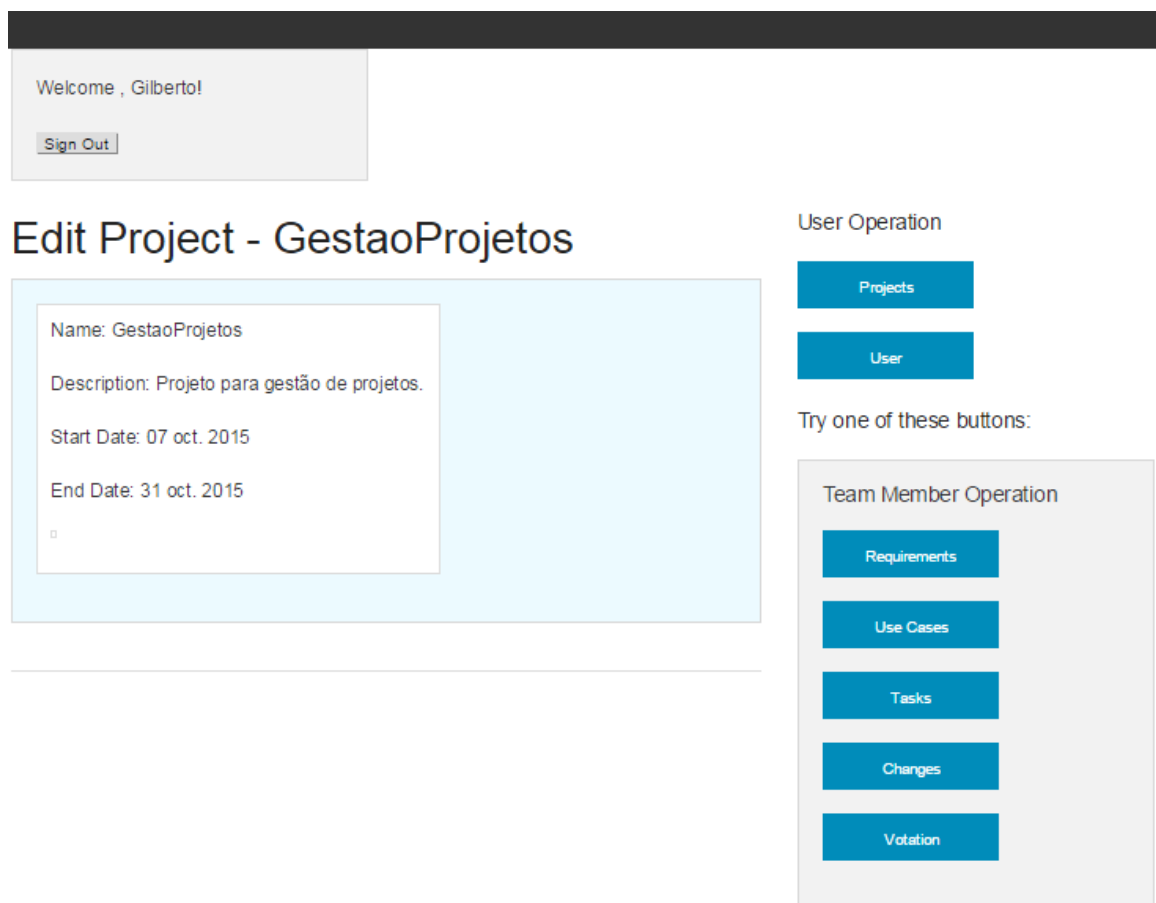


Figura 30 - Team Member – Operações do Team Member.

A Figura 30 apresenta as funcionalidades que um *Team Member* tem ao seu dispor, ao longo do desenvolvimento do projeto. Em comparação com o *Project Manager*, o *Team Member* não assume a responsabilidade de construir a equipa de análise e de projeto, podendo no entanto contribuir tecnicamente no desenvolvimento do mesmo através das outras operações de que dispõe e lhe são familiares.

De acordo com o modelo PMBOK, a equipa de projeto pode agrupar outros intervenientes que não sejam particularmente preparados tecnicamente, mas que em termos de conhecimentos práticos daquilo que se pretende obter, são bastante lúcidos apesar de muitas vezes não o conseguirem expressar de forma uniforme.

De forma a promover a mesma linguagem, à medida que o projeto avança, a gestão de recursos é uma realidade, sendo importantíssimo saber escutar ou promover meios de interação entre os intervenientes de grau técnico diferente.

Neste sentido a aplicação dispõe das operações necessárias a suprir essas dificuldades, habituais na gestão de projetos e ressalvadas no modelo PMBOK. Os *Other Stakeholders* têm ao seu dispor um conjunto de operações que lhes permite dar o seu contributo partilhado no desenvolvimento do projeto.

The screenshot displays a web application interface for editing a project. At the top, a dark header bar is visible. Below it, a light gray box contains the text 'Welcome , andre!' and a 'Sign Out' button. The main content area is titled 'Edit Project - GestaoProjetos'. On the left, a light blue box contains a form with the following fields: 'Name: GestaoProjetos', 'Description: Projeto para gestão de projetos.', 'Start Date: 07 oct. 2015', and 'End Date: 31 oct. 2015'. To the right of the form, there is a sidebar. At the top of the sidebar, it says 'User Operation' with two buttons: 'Projects' and 'User'. Below this, it says 'Try one of these buttons:' followed by a box titled 'Stakeholder Operations' containing three buttons: 'Requirements', 'Changes', and 'Votation'.

**Figura 31 - Stakeholder – Operações do Stakeholder.**

A Figura 17 ilustra quais as funcionalidades que o *Stakeholder* tem ao seu dispor de forma a participar ativamente no desenrolar do projeto. Note-se que neste caso prático o *Stakeholder* foi escolhido para fazer parte da equipa de análise, por isso tem a responsabilidade acrescida de participar em qualquer alteração de requisitos que achar

interessante ressaltar, bem como, contrapor outras mudanças submetidas por outros intervenientes.

Em termos gerais a aplicação tenta responder aos requisitos definidos no modelo PMBOK, com uma plataforma web centrada na gestão de *Stakeholders*, gestão da comunicação e gestão de requisitos.

#### 5.4 CONCLUSÕES OU NOTAS FINAIS

Neste capítulo foi apresentada a aplicação desenvolvida, ilustrando os perfis de utilização da mesma, e em especial a gestão de *Stakeholders* e de recursos no projeto.

Foi apresentada a estrutura seguida na construção da aplicação web desenvolvida, bem como a organização prática de toda a informação ao nível da gestão de projetos. O arrumamento da aplicação tem por base o modelo PMBOK, pelo que foram especificados os comportamentos que cada *Stakeholder* deve ter num projeto de gestão através das suas intervenções colaborativas.

Foi, ainda apresentado, um caso prático, demonstrando os diferentes papéis que um *Stakeholder* pode ter num projeto e de que maneira contribuem para o desenvolvimento do projeto. No caso concreto foram apresentadas as operações comportamentais que a ferramenta reserva para cada perfil do utilizador.



## 6. CONCLUSÕES E TRABALHO FUTURO

### 6.1 INTRODUÇÃO

O sucesso da gestão de projetos depende em grande medida da qualidade analítica e rigor imposto em cada etapa do ciclo de vida do projeto.

O PMBOK ajuda a que a qualidade da solução possa ser avaliada em todas as suas fases (Iniciação, Planeamento, Monitorização e Controlo, Execução e Encerramento), permitindo que os problemas detetados possam ser analisados atempadamente.

O objetivo deste projeto de mestrado consistiu na criação de uma ferramenta de suporte à gestão de projetos de software, que fosse capaz de interpretar, monitorizar e gerir de uma forma comum as necessidades do cliente. Não se pretendia apresentar uma ferramenta final, definitiva, mas sim elaborar todo o processo de análise de sistema e produção de artefactos de engenharia de software, assim como construir um incremento de software já com alguma complexidade e que permitisse testar e levantar mais requisitos com os utilizadores em contacto com a ferramenta.

A ferramenta apresentada neste projeto, permite a implementação de conceitos de gestão de projetos baseados no modelo PMBOK, com o utilizador a um alto nível de abstração, independentemente da linguagem de programação utilizada no desenvolvimento da aplicação e do modelo definido na sua implementação. Assim, a ferramenta desenvolvida, representa uma abordagem ao PMBOK com interfaces gráficas, que permitem ao utilizador final interagir de uma forma guiada e contextualizada no modelo do projeto.

## 6.2 CONTRIBUTO

O principal contributo deste trabalho é, tal como referido acima, a ferramenta de gestão de projetos Open Source. A ferramenta foi construída usando a versão mais recente de frameworks de desenvolvimento web, e foi desenvolvida por forma a ser expansível e configurável. Mesmo não tendo por objetivo a criação de uma aplicação final completa de um dado software, tem implementados vários conceitos abordados na gestão de projetos no modelo PMBOK, nomeadamente a gestão de *Stakeholder* e gestão de recursos.

Os capítulos 4 e 5 apresentaram a comparação de diferentes ferramentas de gestão de projetos, a estrutura da ferramenta e um exemplo de aplicação respetivamente, da aplicação desenvolvida.

No capítulo 3 foi abordado o estado da arte. Este estudo teve por base como vem sido referido, o PMBOK, um modelo de referência na gestão de projetos. Este estudo foi efetuado de forma a importar os conceitos básicos de gestão no desenvolvimento de software.

## 6.3 DISCUSSÃO DE RESULTADOS

Como referido em capítulos anteriores, a ferramenta apresentada surge no seguimento de outras ferramentas Open Source já existentes na área de gestão de projetos. Atente-se ao capítulo 3.2, o qual apresenta uma análise comparativa de diferentes ferramentas alternativas para a gestão de projetos de software.

Os critérios de comparação foram selecionados de forma a serem analisados aspetos conceptuais, funcionais e aspetos relacionados com os objetivos definidos pelo projeto.

A opção de selecionar várias ferramentas deveu-se ao facto de cada uma delas ter objetivos similares, embora os mesmos sejam abordados de uma forma diferente.

Nesta secção, foi selecionada uma dessas ferramentas para comparação de resultados com a ferramenta desenvolvida. Selecionou-se a ferramenta “Open Project Foundation”, por ser a mais semelhante à ferramenta desenvolvida e aqui apresentada. Com efeito nas outras ferramentas apresentadas no capítulo 3.2, é dada a relevância a aspetos comportados pela ferramenta em análise mas que por outro lado sublimam outros tão mais importantes como a equipa de análise do projeto.

**Tabela 2 - Tabela comparativa entre o Open Project Foundation e a aplicação desenvolvida.**

	Open Project Foundation	Aplicação desenvolvida
<b>Ambiente de suporte</b>	Web	Web
<b>Licença</b>	Gratuita	Gratuita
<b>Gestão de requisitos</b>	A gestão de requisitos é limitada à criação de tarefas atribuídas a cada utilizador.	A gestão de requisitos conhece a distinção entre diferentes níveis analíticos da gestão de requisitos. O utilizador dispõe automaticamente da gestão de responsabilidades, podendo criar casos de uso, tarefas e requisitos a eles associados.
<b>Gestão da comunicação</b>	A ferramenta considera a comunicação entre os intervenientes de uma forma geral, não fazendo a distinção de responsabilidades.	A gestão da comunicação é uma realidade através da atribuição e distinção de responsabilidades entre os diferentes membros do projeto

<b>Gestão de Stakeholders</b>	A gestão de <i>Stakeholders</i> não é uma realidade prática. Todos os utilizadores são intervenientes do projeto e partilham informação, não havendo a sensibilidade adequada a cada interveniente.	Todos os intervenientes têm um papel no projeto, distinguido na aplicação. Com o alinhamento da informação comum todos os utilizadores sentem abertura de dar o seu contributo.
<b>Calendarização</b>	É feita a calendarização por tarefas a cada utilizador dentro do projeto.	A calendarização é centrada nos recursos humanos e nas tarefas a eles atribuídos, gerando um grau de prioridade entre as tarefas.
<b>Gestão de equipas de análise</b>	A ferramenta é centrada no utilizador onde todos partilham o conhecimento na ferramenta.	A equipa de análise é nomeada pelo chefe projeto, de forma a dar o contributo específico e com precisão, mas especializado no domínio.
<b>Gestão da qualidade e riscos</b>	A gestão da qualidade e de riscos é centrada na calendarização das tarefas no projeto.	A gestão da qualidade e de riscos é centrada na calendarização dos vários requisitos e da equipa de análise nas suas tomadas de decisão.

## 6.4 PRINCIPAIS RESULTADOS OBTIDOS

Tendo em consideração os objetivos inicialmente propostos, e para concluir a apresentação do trabalho realizado, podemos destacar a importância dos modelos de gestão de projetos no desenvolvimento de ferramentas ao seu suporte.

Com o desenvolvimento desta ferramenta podemos destacar as seguintes características:

- Implementação numa versão mais recente da tecnologia, a gestão de projetos de software.



- A interpretação e monitorização das necessidades dos clientes nos projetos em desenvolvimento: com a participação de cada *Stakeholder* no projeto de uma forma controlada.
- A visualização de diferentes níveis de informação, de acordo com a responsabilidade de cada interveniente no projeto em desenvolvimento.
- A possibilidade de nomear uma equipa de projeto que trabalha em prol do objetivo final, definido como âmbito neste trabalho.
- A implementação prática do conceito de equipa de análise, que permite delegar responsabilidades e selecionar os interlocutores com poder de decisão para analisar pedidos de alteração ao projeto. A gestão de alterações baseia-se no conceito de votação. Neste sentido todos os intervenientes da equipa de análise poderão dar o seu ponto de vista ao gestor de projeto de forma a abrir entendimentos e consensos, contribuindo também para aprendizagem para projetos futuros.
- A introdução do conceito de informação partilhada, através da plataforma. Com recurso ao desenho e implementação de uma aplicação que congrega a monitorização e o controlo de todos os requisitos.
- Alinhamento dos requisitos definidos com modelos gráficos que favoreçam o desenvolvimento ágil mas baseado em modelos.
- Acompanhamento durante o desenvolvimento e em projetos futuros com base em projetos anteriormente realizados.

Toda a ferramenta comporta uma base comum, permitir a todos os intervenientes, participarem no desenvolvimento do projeto. Com esta ferramenta foram apresentados em termos práticos alguma dessa gestão e que permite reduzir em muito, o insucesso do desenvolvimento dos projetos de software.

## 6.5 TRABALHO FUTURO

Um dos objetivos iniciais na estrutura desta ferramenta foi permitir que esta pudesse ser genérica e expansível. Tendo em consideração o trabalho descrito, bem como alguns aspetos que foram sendo definidos ao longo do trabalho. Existe ainda um conjunto de funcionalidades que poderão fazer parte desta ferramenta, tornando-a mais completa, das quais podemos salientar:

- Implementação da gestão de históricos, que permite dar o suporte necessário à aprendizagem e otimização de processos de gestão, sejam eles ao nível dos requisitos propriamente ditos, como os resultados finais obtidos.
- Geração de modelos gráficos que permita uma análise mais concreta de todos os intervenientes. Neste sentido fica para trabalho futuro a implementação de importações de vários tipos de ficheiros no suporte à análise de requisitos e definição de Storyboards.
- A criação e implementação de atividades e sua precedência, no sentido de apoiar o detalhe dos requisitos.
- A monitorização de entidades de domínio bem como os modelos que as suportam.

Apesar da execução da lista de tarefas identificada acima, apresentar um conjunto de operações ainda a desenvolver para a obtenção de uma ferramenta ainda mais completa e útil, a versão atual já funciona e apresenta conforme tem sido anunciado nos capítulos anteriores a implementação práticas de muitos desses conceitos.

Seria extremamente interessante num estudo próximo, que se realizasse a manutenção e gestão de todo o conteúdo concetual desenvolvido, uma vez que este se encontra devidamente analisado com possibilidades de implementação prática.



## REFERÊNCIAS

- [1] Campozano, Nelson (2013). As camadas MVC  
Local: <http://fabrica.ms.senac.br/2013/06/as-camadas-mvc/>
- [2] PMBOK Guide, third, fourth and fifth edition (2013). A Guide to the project management body of knowledge
- [3] Scott, Bill (2012). What Interface Engineers wish Designers Knew - Interview
- [4] PMI, (2012). Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK), Quinta Edição em Português.  
Project Management Institute (PMI). Global Standard, dezembro 2012, EUA. ISBN: 978-1-62825-007-7
- [5] Philips, Joseph (2002). The PMP Lab Manual.
- [6] Philips, Joseph (2002). Project Management Professional Study Guide
- [7] Philips, Joseph (2002). Project Management for Small Business
- [8] Clark, (1999). Towards a cognitive robotics
- [9] Gonçalves, António (2013). Java 7 for Absolute Beginners. Editora: Apress
- [10] Rocha, Paulo (2012). Análise da adoção do Open Source nos municípios portugueses. Dissertação de mestrado, Faculdade de Economia, Univ. do Porto.
- [11] Barbosa, Fernando (2013). Gerenciamento de Projetos PMBOX.  
Local: <http://w3.ufsm.br/proplan/images/stories/file/COPLIN/PMBOK-UFSM-Aula01.pdf>
- [12] Trentim, Mario (2015). Os 47 processos do guia PMBOK 5ª edição.  
Revista MundoPM.  
Local: <http://www.hmdoctors.com/index.php/2013/04/areas-de-conhecimento-do-gerenciamento-de-projetos>

## REFERÊNCIAS WWW

- [13] Arquitetura MVC:  
<https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>
- [14] Spring Scaffolding Support in MyEclipse:  
<https://www.genuitec.com/products/myeclipse/learning-center/spring/overview-of-myeclipse-with-spring-support>
- [15] Fatores críticos de sucesso na gestão de projetos na perspectiva da gestão do conhecimento:  
<https://repositorio.uninove.br/xmlui/bitstream/handle/123456789/440/545-978-1-RV%20-%20fatores%20criticos%20de%20sucesso%20na%20gp.pdf?sequence=1>
- [16] Gerenciamento de Projetos Web, 10 Causas de Fracasso:  
<http://fr.slideshare.net/s3k7or/principais-causas-de-fracasso-em-um-projeto>
- [17] Quais as áreas de conhecimento mais importantes:  
<http://blog.mundopm.com.br/2012/01/25/qual-a-area-do-conhecimento-mais-importante-em-gerenciamento-de-projetos/>
- [18] Áreas de conhecimento do PMBOX:  
<http://www.portal-administracao.com/2014/06/areas-do-conhecimento-guia-pmbok.html>
- [19] Habilidades Interpessoais:  
<http://escritoriodeprojetos.com.br/habilidades-interpessoais.aspx>
- [20] PMBOX Guide 5ª edição – O papel do gerenciamento do projeto:  
<http://www.vic.ms/project-management/pmbok-guide-5a-edicao-secao-1-7-o-papel-do-gerente-de-projeto/>
- [21] The everything practice interview book:  
<http://www.ssnpstudents.com/wp/wp-content/uploads/2015/01/The-Everything-Practice-Interview-Boo.pdf>
- [22] O uso de Software Open Source nos Sistemas de Informação de Empresas:  
<http://www.neoscopio.com/docs/foss/egp-foss-si.pdf>
- [23] Spring Security in Servlet Web Application using Dao:  
<http://www.journaldev.com/2715/spring-security-in-servlet-web-application-using-dao-jdbc-in-memory-authentication>
- [24] Spring Guide:  
<https://spring.io/guides/gs/securing-web/>

- [25] Spring MVC Session:  
<http://www.javacodegeeks.com/2013/04/spring-mvc-session-tutorial.html>
- [26] Spring Security:  
<http://javahash.com/spring-security-hello-world-example/>
- [27] Html Template :  
<http://foundation.zurb.com/templates.html>
- [28] Thymeleaf documentation :  
<http://www.thymeleaf.org/documentation.html>
- [29] Beginner's Book:  
<http://beginnersbook.com/java-tutorial-for-beginners-with-examples/>
- [30] Spring Context – Maven Repository :  
<http://mvnrepository.com/artifact/org.springframework/spring-context>
- [31] Análise de Requirement de Software:  
[https://pt.wikipedia.org/wiki/An%C3%A1lise\\_de\\_requerimento\\_de\\_software](https://pt.wikipedia.org/wiki/An%C3%A1lise_de_requerimento_de_software)
- [32] PMBOX – Criar estrutura analítica do projeto. (EAP) :  
<http://pm2all.blogspot.pt/2011/09/pmbok-53-criar-estrutura-analitica.html>
- [33] Discover an Open Source:  
<http://opensource.com/resources/what-open-source>
- [34] O que é Open Source:  
<http://canaltech.com.br/o-que-e/o-que-e/O-que-e-open-source/>
- [35] Caso de estudo de sucesso da implementação open source, (2011):  
[http://wiki.ua.sapo.pt/wiki/COS\\_Repensar\\_o\\_Open\\_source\\_Caso\\_de\\_Estudo](http://wiki.ua.sapo.pt/wiki/COS_Repensar_o_Open_source_Caso_de_Estudo)
- [36] Estudo de poupança com Open Source, (2014):  
<http://www.computerworld.com.pt/2014/09/02/estudo-confirma-potencial-de-poupanca-com-open-source/>
- [37] Open source Software – Que oportunidades em Portugal:  
[http://www.algebraica.pt/i\\_ap/bo2/data/upimages/Estudo\\_Open\\_Source\\_com\\_capa.pdf](http://www.algebraica.pt/i_ap/bo2/data/upimages/Estudo_Open_Source_com_capa.pdf)
- [38] Padrões DAO:  
[http://www.macoratti.net/11/10/pp\\_dao1.htm](http://www.macoratti.net/11/10/pp_dao1.htm)
- [39] Rosenfeld Louis, Future Practice Interview:  
[http://www.macoratti.net/11/10/pp\\_dao1.htm](http://www.macoratti.net/11/10/pp_dao1.htm)





# ANEXOS

## ANEXO A: STORYBOARDS

### A.1 STORYBOARD DO PROJECT MANAGER

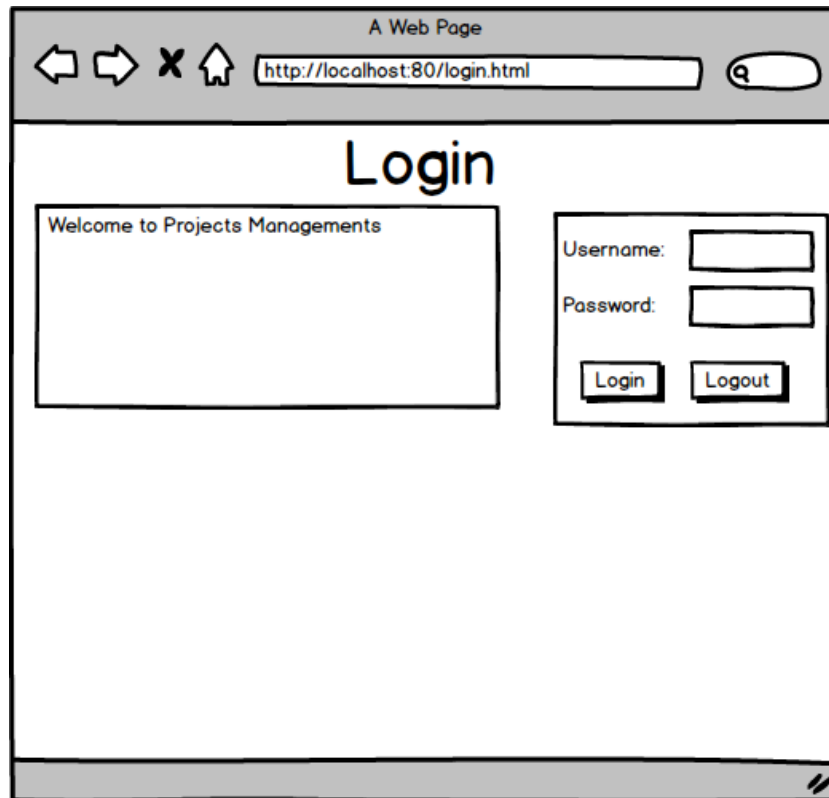
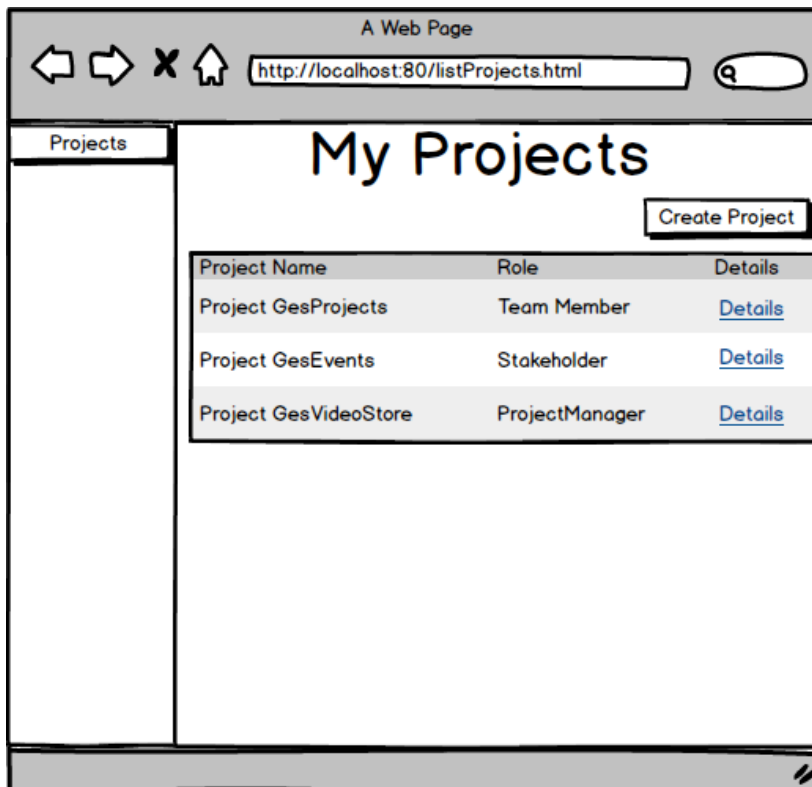
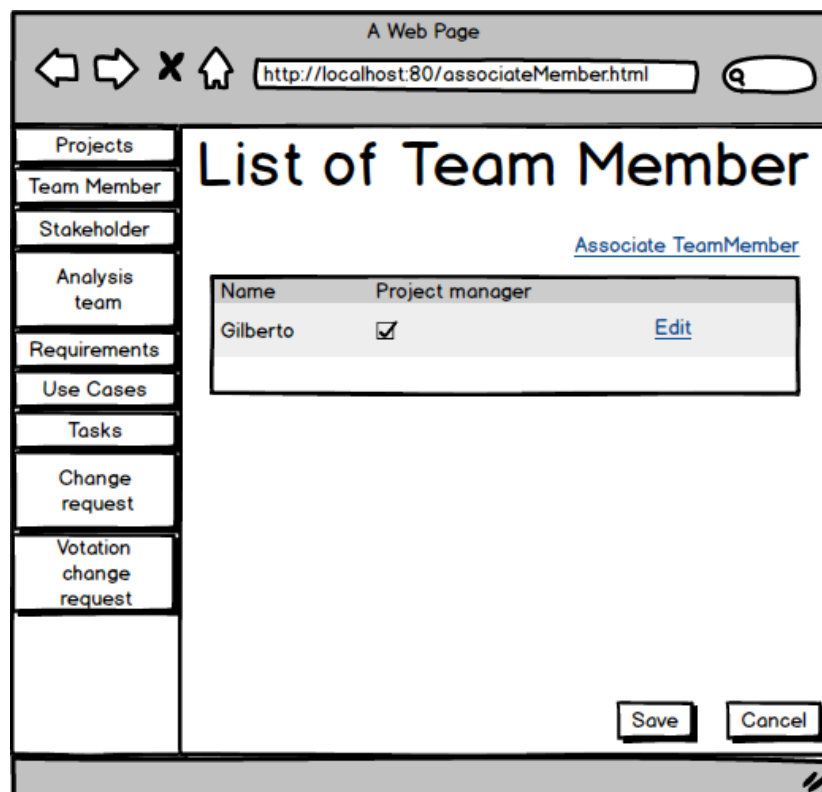


Figura 32 - Project Manager – Login.



It should select the project that we want to work.

Figura 33 - Project Manager - Meus Projetos.



if we click for details, the window team member will appear showing the team member features in the project.

Figura 34 - Project Manager - Lista dos Team Members.

A Web Page

http://localhost:80/teamMember.html

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

## Team Member

User: Gilberto ▾ ☐ Project manager

All Skills:

analyst  
programmer

Team member's Skills:

analyst

>

<

Save Cancel

Figura 35 - Project Manager - Team Member.

A Web Page

http://localhost:80/associateMember.html

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

## List of stakeholder

[Associate Stakeholder](#)

Name
Miguel <a href="#">Edit</a>

Save Cancel

if we click for details, the window Stakeholder will appear showing the stakeholder's features in the project.

Figura 36 - Project Manager - Lista de Stakeholders.

A Web Page

http://localhost:80/stakeholder.html

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

# Stakeholder

User: Miguel ▼ Role:  Add

Roles Associated:

operational assistant

Save Cancel

Figura 37 - Project Manager - Stakeholder.

A Web Page

http://localhost:80/analysisTeam.html

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

# Analysis Team

TeamMembers

Gilberto  
Ana

Analysis Team Members

Gilberto

Stakeholders

Joana  
Miguel

Analysis Team Members

Miguel

Save Cancel

Figura 38 - Project Manager - Equipa de Análise.

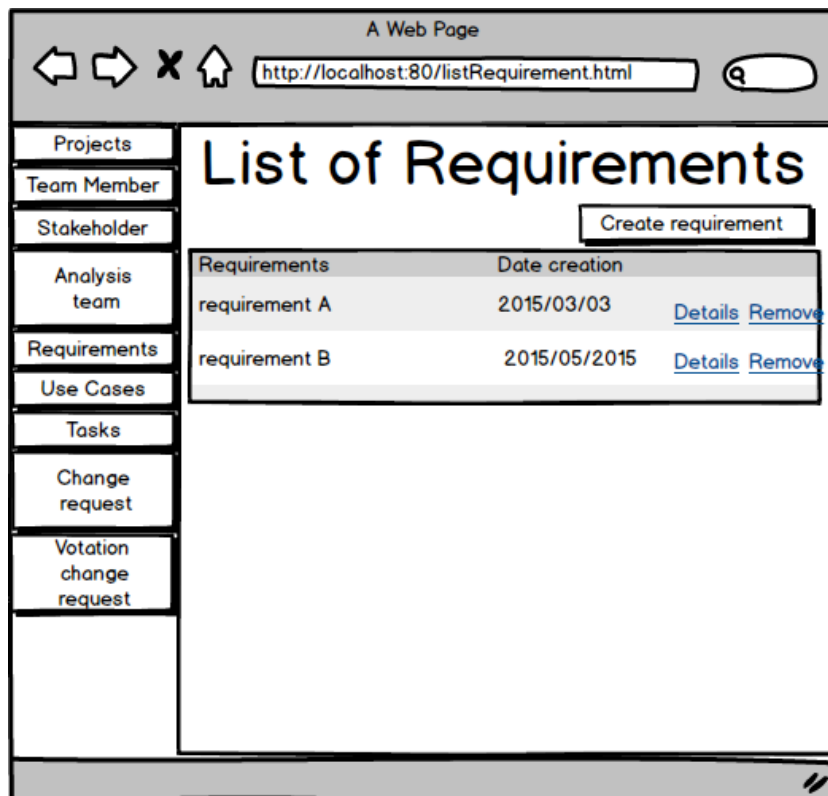


Figura 39 - Project Manager - Lista de requisitos.

A Web Page
http://localhost:80/listRequirement.html

Projects
Team Member
Stakeholder
Analysis team
Requirements

# Requirements

Name: requirement B

Description: description of requirement B

Status: created ▼

Degree importance: 1

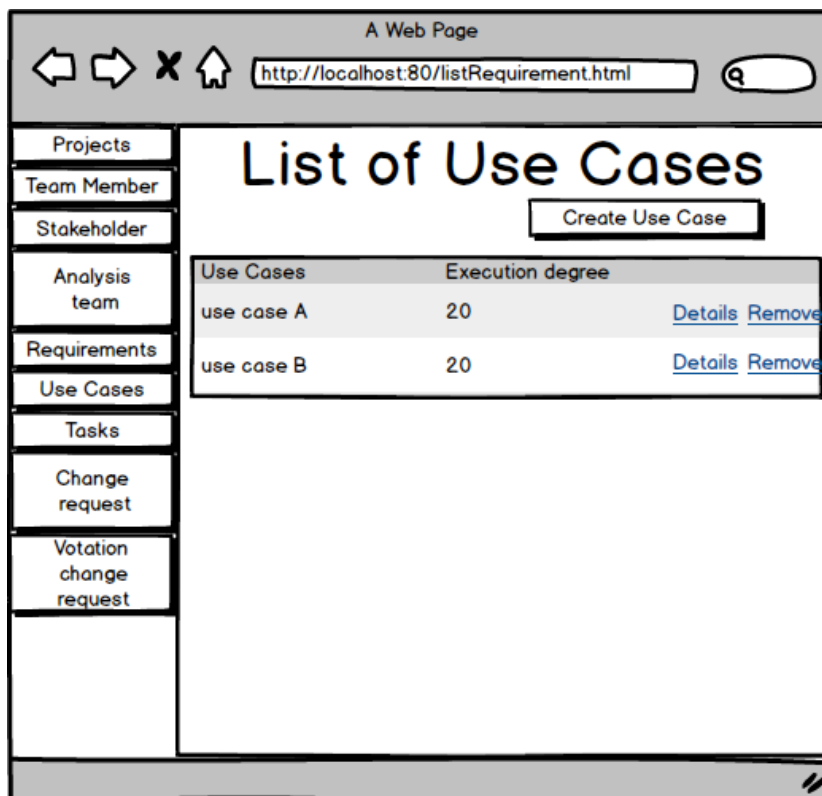
Save Cancel

Associated use cases [associate use case](#)

Use Case	
use case B	<a href="#">Details</a>

Associate Use Case allows to associate the use cases to the requirement

Figura 40 - Project Manager - Requisito.



Use cases list that we are working. Link Details allows to going use case selected.

Figura 41 - Project Manager - Lista de casos de uso.

A Web Page
http://localhost:80/useCase.html

Projects
Team Member
Stakeholder
Analysis team
Requirements
Use Cases
Tasks
Change request
Votation change request

# Use Case

Name: use case B

Description: description of use case B

Estimated time: 3 days

ExecDegree: 20 %

Save Cancel

Associated tasks [associate tasks](#)

Tasks	Duration	
task B	30	<a href="#">Details</a>

Associated requirements [associate requirements](#)

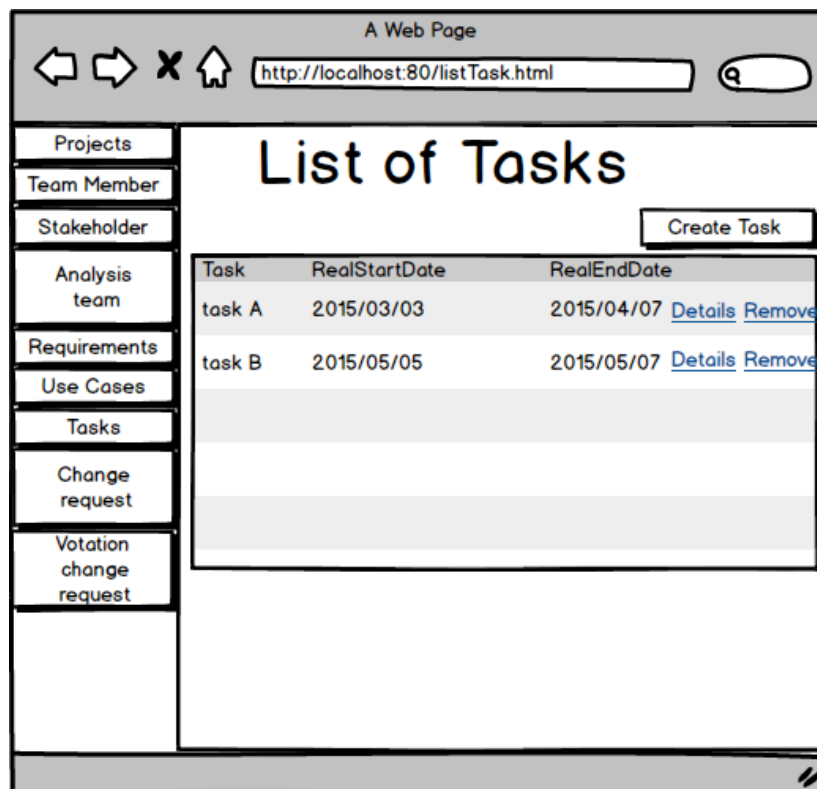
Requirements	Duration	
requirement B	30	<a href="#">Details</a>

Associate task allows to associate existing tasks to the use case on the project we are working.

Associate requirement allows to associate existing requirements to the use case on the

Figura 42 - Project Manager - Caso de uso.





Task list that we are working. Link Details allows to going task selected.

Figura 43 - Project Manager - Lista de tarefas.

A Web Page
http://localhost:80/task.html

Projects
Team Member
Stakeholder
Analysis team
Requirements
Use Cases
Tasks
Change request
Votation change request

# Task

Name:

Description:

Duration:

Estimated start date:

Estimated end date:

Real start date:

Real end date:

ExecDegree: 20 %

Associated use cases [associate use case](#)

Use Case	
use case B	<a href="#">Details</a>

Associate Use Case allows to associate the use cases to the task being edited.

Figura 44 - Project Manager - Tarefa.

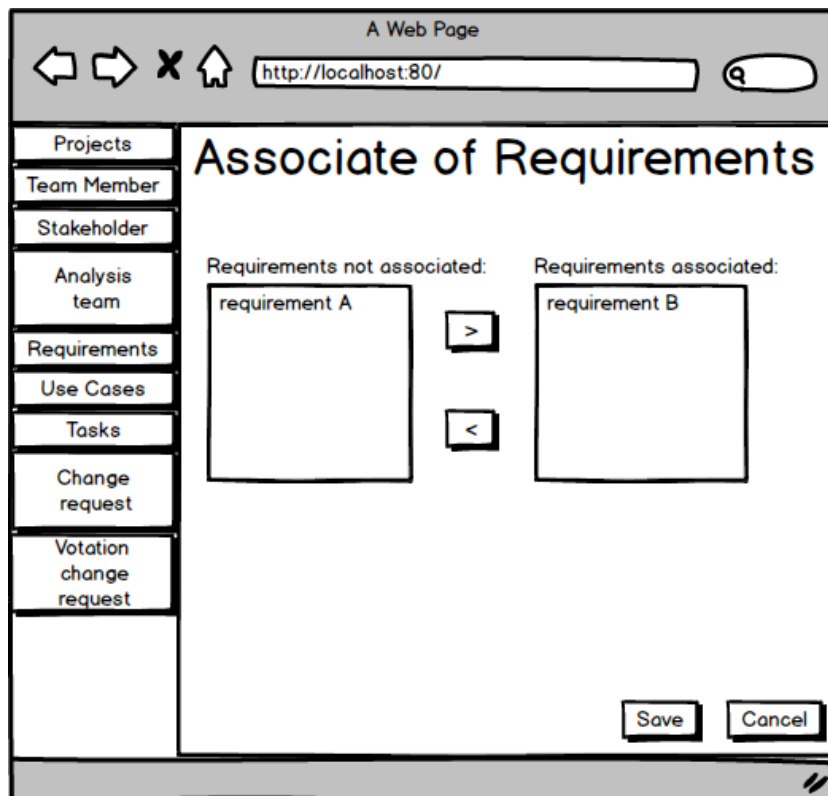


Figura 45 - Project Manager - Associação de requisitos.

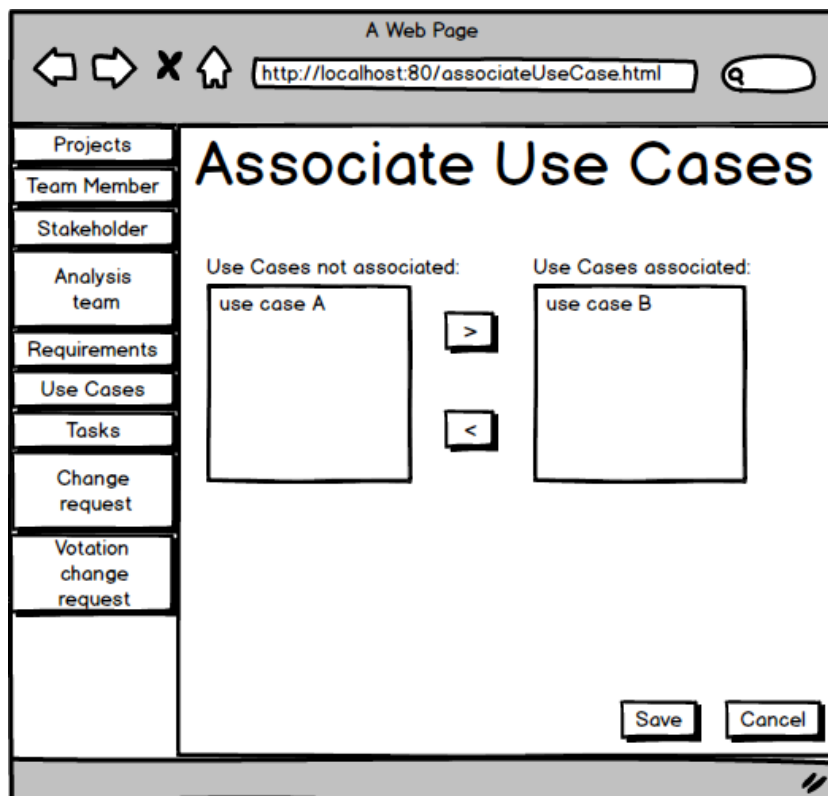


Figura 46 - Project Manager - Associação de casos de uso.

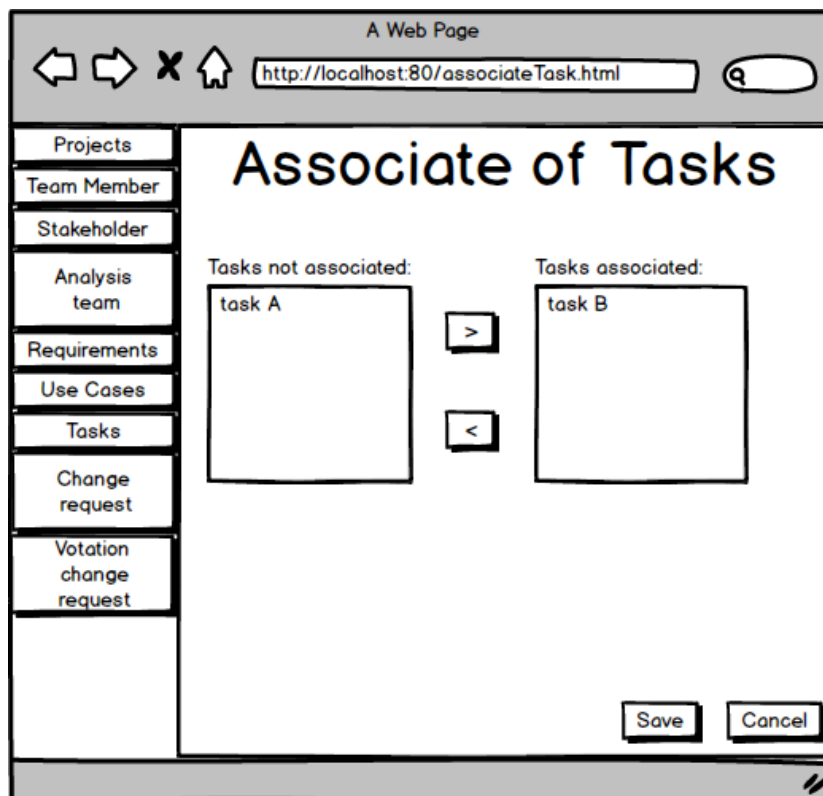
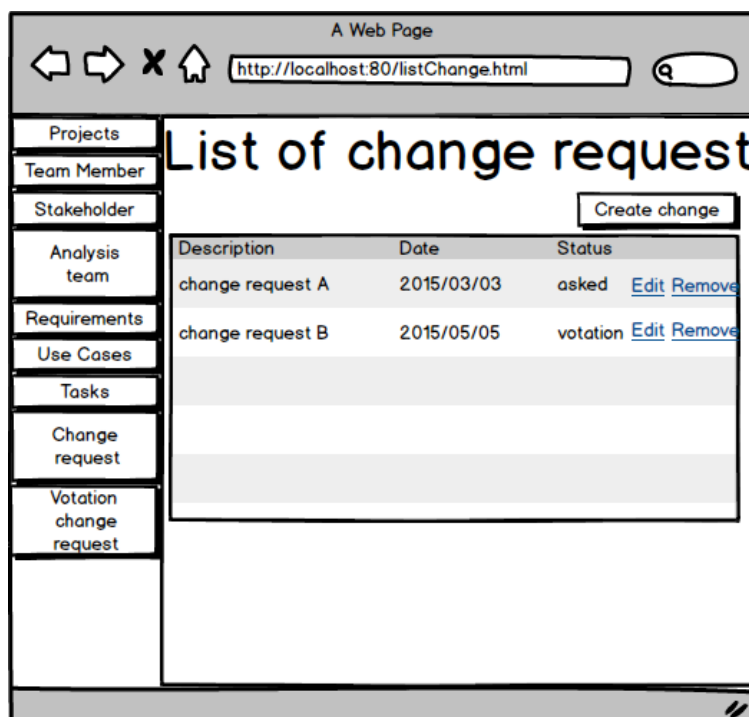


Figura 47 - Project Manager - Associação de tarefas.



Change request list that we are working. Link Edit allows to going edit change request selected.

Figura 48 - Project Manager - Lista de pedidos de alteração.

A Web Page

http://localhost:80/changeRequest.html

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

## Change Request

Description: Proposing new end date of project and changing project requirement

Requirement: requirement B ▼

Propose new End date: 2015/12/12

Status: asked ▼

Save Cancel

Figura 49 - Project Manager – Pedido de alteração.

A Web Page

http://localhost:80/listVotationRequest.html

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

## List of votation

Description	Date	Status	
change request B	05/05/2015	votation	<a href="#">Vote</a>

Votation request list that we are working. Link Vote allows to going vote on the change request selected.

Figura 50 - Project Manager - Lista de votações.

A Web Page

⬅ ➡ ✕ 🏠
http://localhost:80/votationRequest.html
🔍

Projects

Team Member

Stakeholder

Analysis team

Requirements

Use Cases

Tasks

Change request

Votation change request

## Votation Request

Description: 

Proposing new end date of project and changing project requirement

Requirement: 

requirement B ▼

Porpose new End date: 

2015/12/12

📅

Members still voted:

Analysis Team Members	Vote
Gilberto	favor

Members have not voted:

Analysis Team Members	Role
Miguel	Team Member

Vote: 

favor ▼

Justification:

Save

Cancel

Figura 51 - Project Manager – Voto.

A.2 STORYBOARD DO TEAM MEMBER

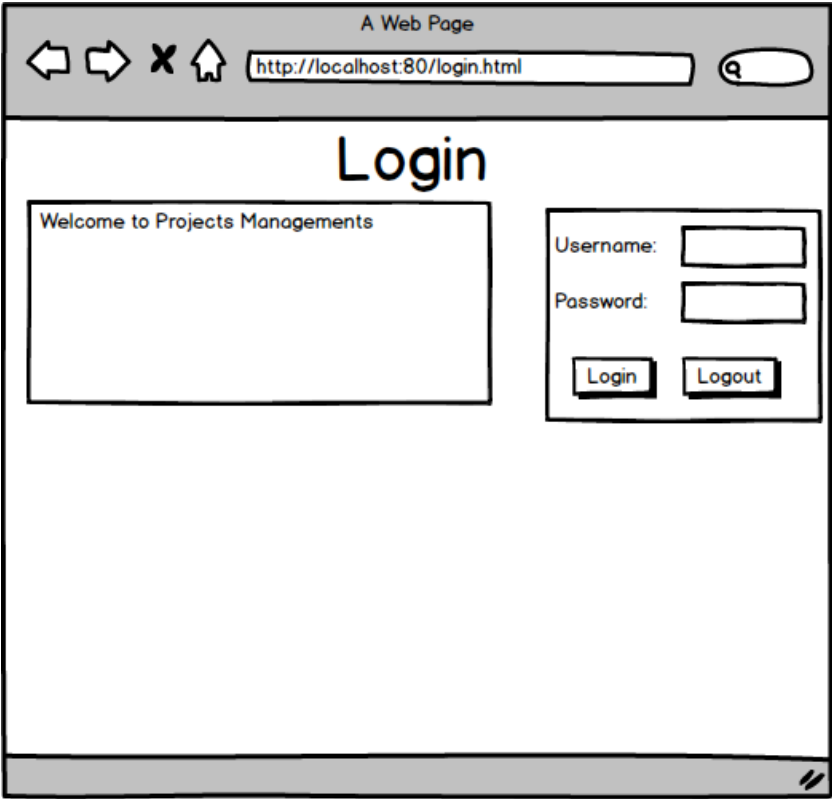


Figura 52 - Team Member - Login.

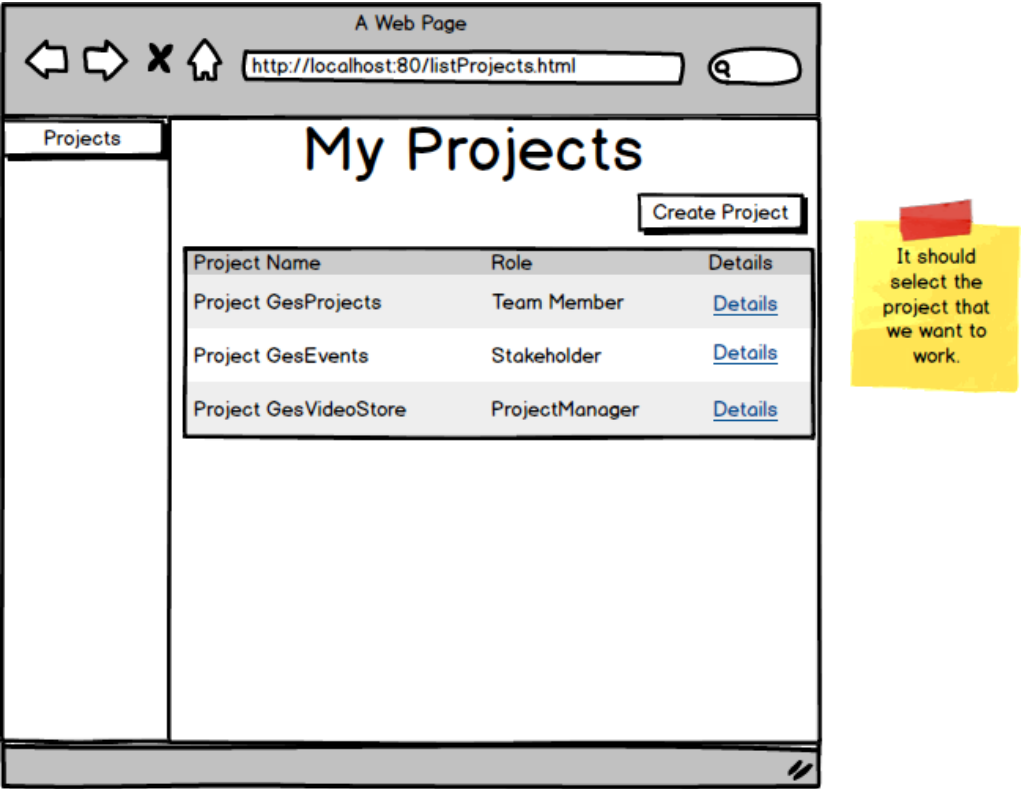
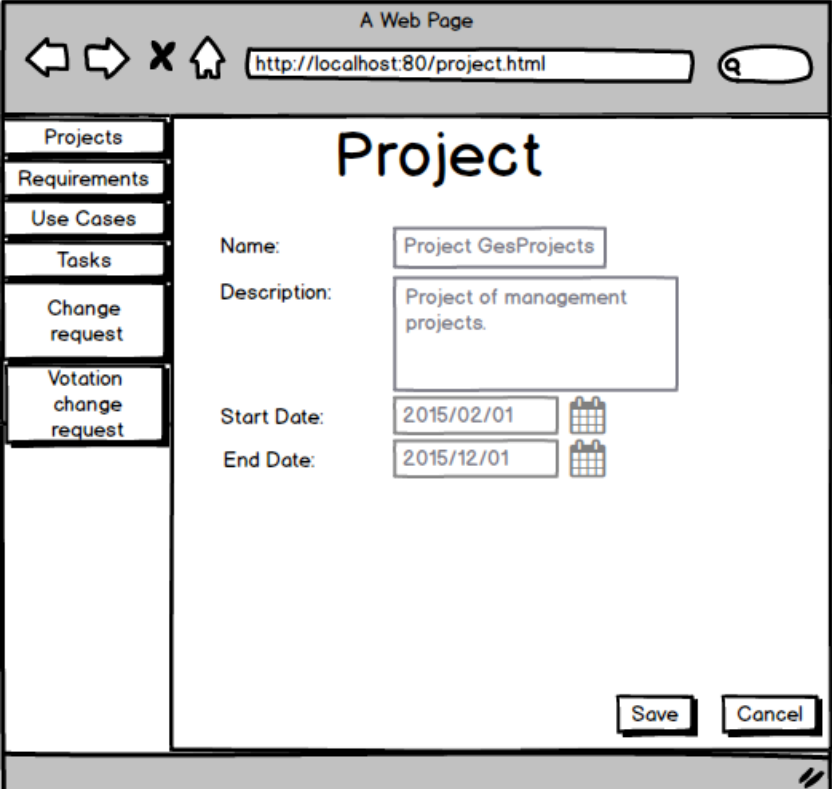


Figura 53 - Team Member - Meus Projetos.



A Web Page

http://localhost:80/project.html

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

## Project

Name: Project GesProjects

Description: Project of management projects.

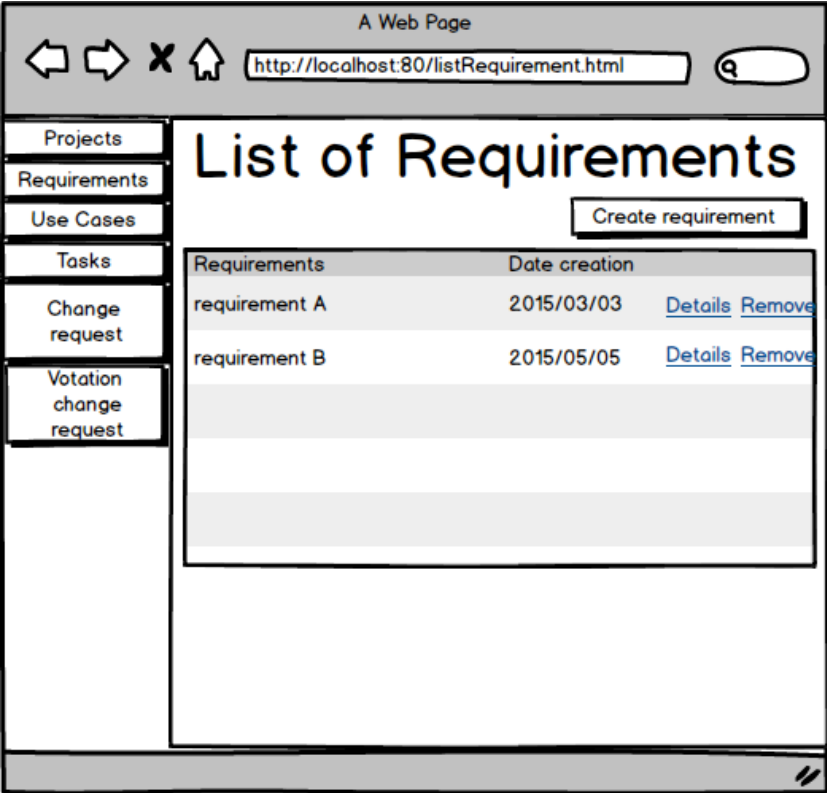
Start Date: 2015/02/01

End Date: 2015/12/01

Save Cancel

List of all fields the selected project.

Figura 54 - Team Member - Projeto.



A Web Page

http://localhost:80/listRequirement.html

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

## List of Requirements

Create requirement

Requirements	Date creation	
requirement A	2015/03/03	<a href="#">Details</a> <a href="#">Remove</a>
requirement B	2015/05/05	<a href="#">Details</a> <a href="#">Remove</a>

Figura 55 - Team Member - Lista de Requisitos.



A Web Page

← → × 🏠  🔍

Projects

Requirements

Use Cases

Tasks

Change request

# Requirements

Name:

Description:

Status:  ▼

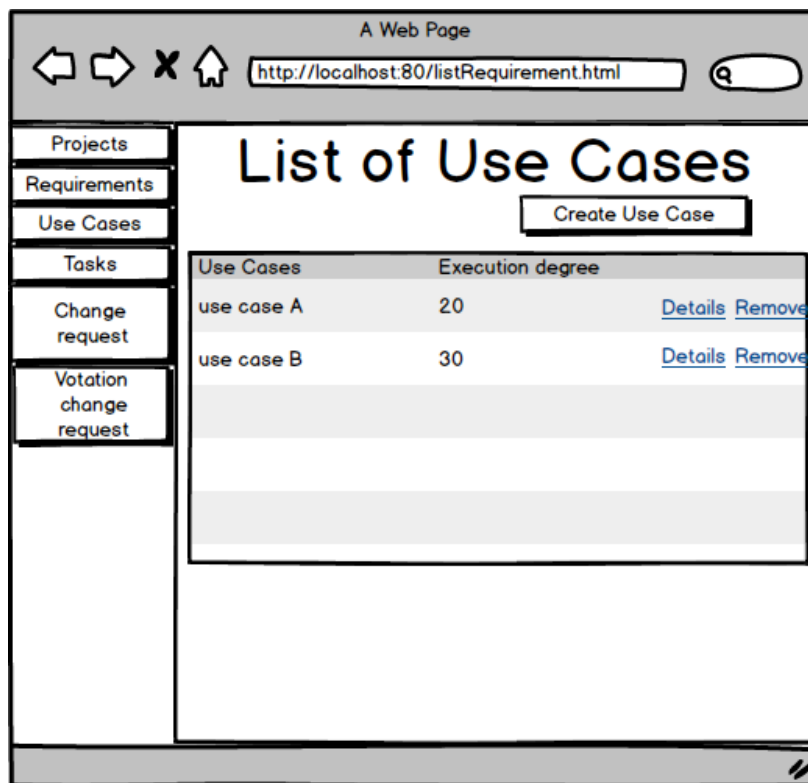
Degree importance:

Associated use cases [associate use case](#)

Use Case
use case B <a href="#">Details</a>

Associate Use Case allows to associate the use cases to the requirement

Figura 56 - Team Member - Requisito.



Use cases list that we are working. Link Details allows to going use case selected.

Figura 57 - Team Member - Lista de casos de uso.

A Web Page

http://localhost:80/useCase.html

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

## Use Case

Name: use case B

Description: description of use case B

Estimated time: 3 days

ExecDegree: 20 %

Save Cancel

Associated tasks [associate tasks](#)

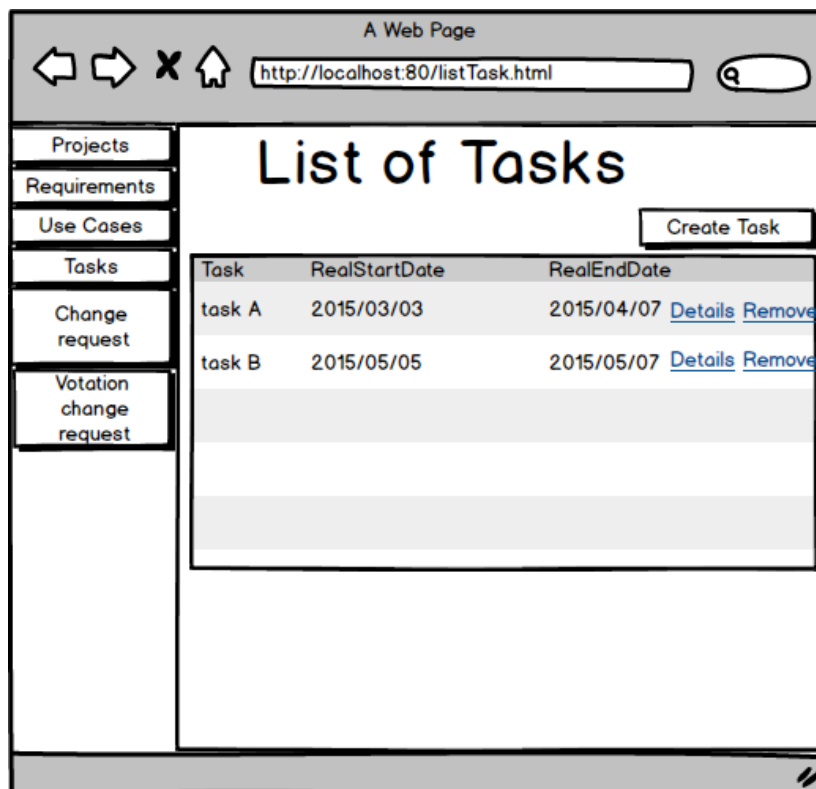
Tasks	Duration	
task B	30	<a href="#">Details</a>

Associated requirements [associate requirements](#)

Requirements	Duration	
requirement B	30	<a href="#">Details</a>

Associate task allows to associate existing tasks to the use case on the project we are working. Create Task allows to create a new task that is automatically associated to the use case.

Figura 58 - Team Member - Caso de uso.



Task list that we are working. Link Details allows to going task selected.

Figura 59 - Team Member - Lista de tarefas.

A Web Page

←

→

✕

🏠

http://localhost:80/task.html

🔍

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

Task

Name:task B

Description:div>

description of task B

Duration:35

Estimated start date:2015/03/03

📅

Estimated end date:2015/04/07

📅

Real start date:2015/03/03

📅

Real end date:2015/04/07

📅

ExecDegree:20 %

Save

Cancel

Associated use cases

[associate use case](#)

Use Case

use case B

Details

Associate Use Case allows to associate the use cases to the task being edited.

Figura 60 - Team Member - Tarefa.

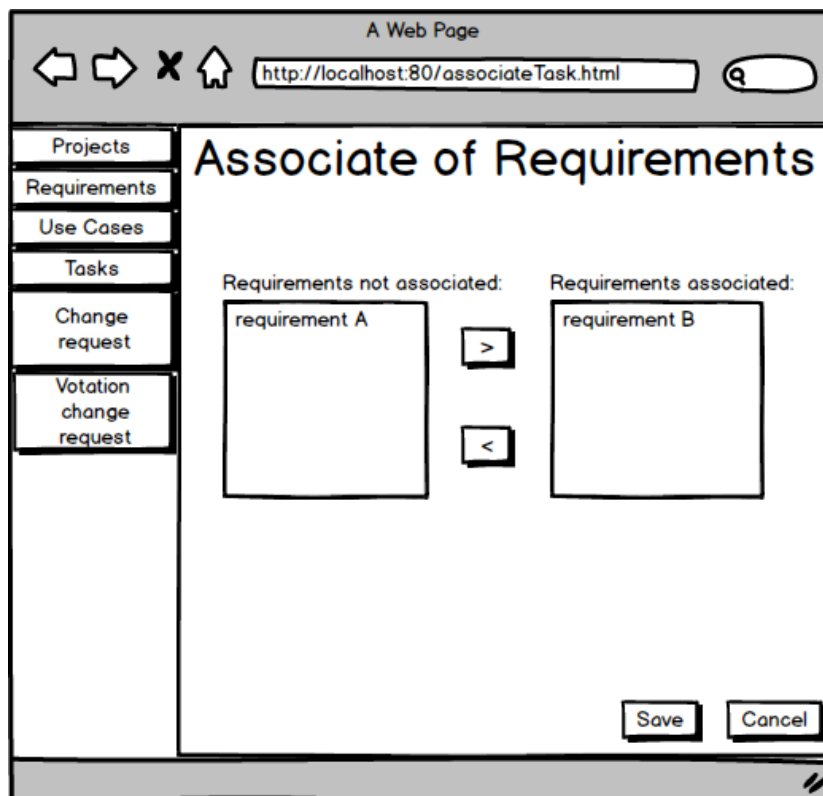


Figura 61 - Team Member - Associação de requisitos.

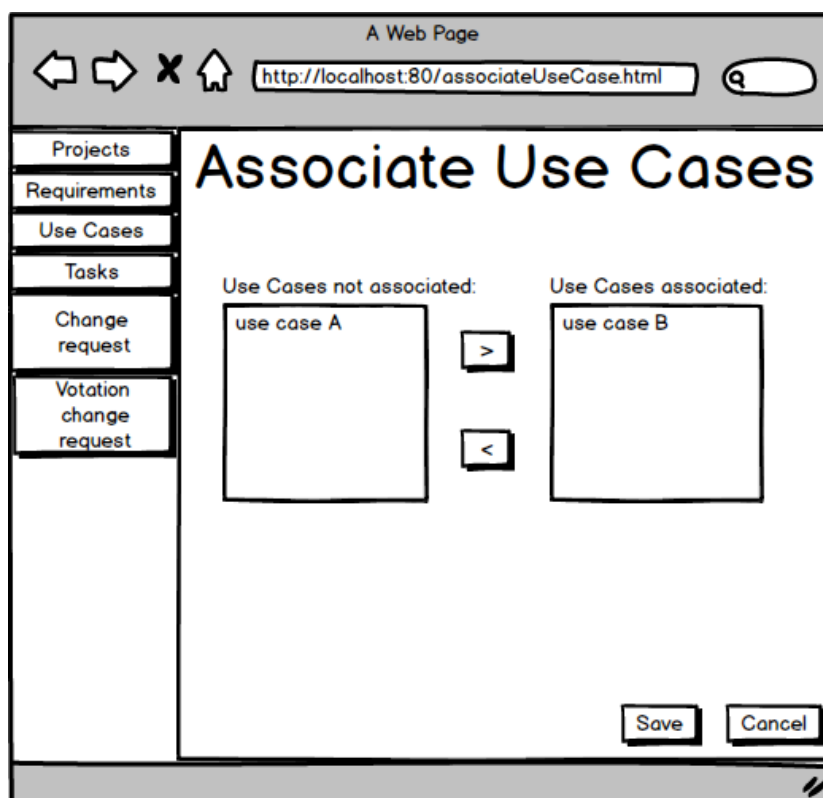


Figura 62 - Team Member - Associação de casos de uso.

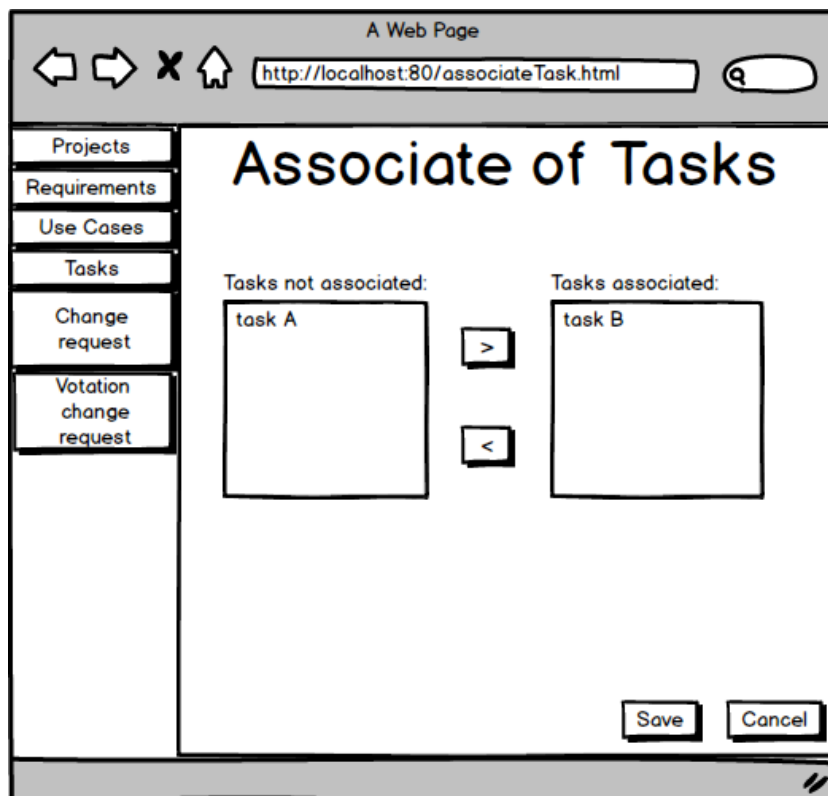
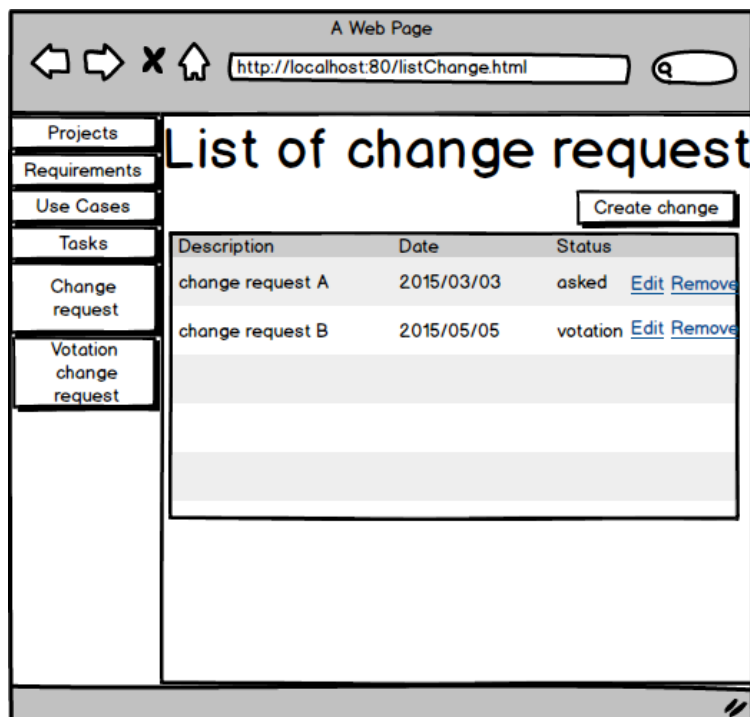


Figura 63 - Team Member - Lista de tarefas.



Change request list that we are working. Link Edit allows to going edit change request selected.

Figura 64 - Team Member - Lista de pedidos de alteração.

A Web Page

http://localhost:80/changeRequest.html

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

## Change Request

Description: Proposing new end date of project and changing project requirement

Requirement: requirement B ▼

Purpose new End date: 2015/12/12

Status: asked ▼

Save Cancel

Figura 65 - Team Member - Pedido de alteração.

A Web Page

http://localhost:80/listVotationRequest.html

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

## List of votation

Description	Date	Status	
change request B	2015/05/05	votation	<a href="#">Vote</a>

The option Votation change request is enable if team member belongs Analysis Team.

Votation request list that we are working. Link Vote allows to going vote on the change request selected.

Figura 66 - Team Member - Lista de votações.



A Web Page

← → × 🏠  🔍

Projects

Requirements

Use Cases

Tasks

Change request

Votation change request

## Votation Request

Description:

Requirement:  ▼

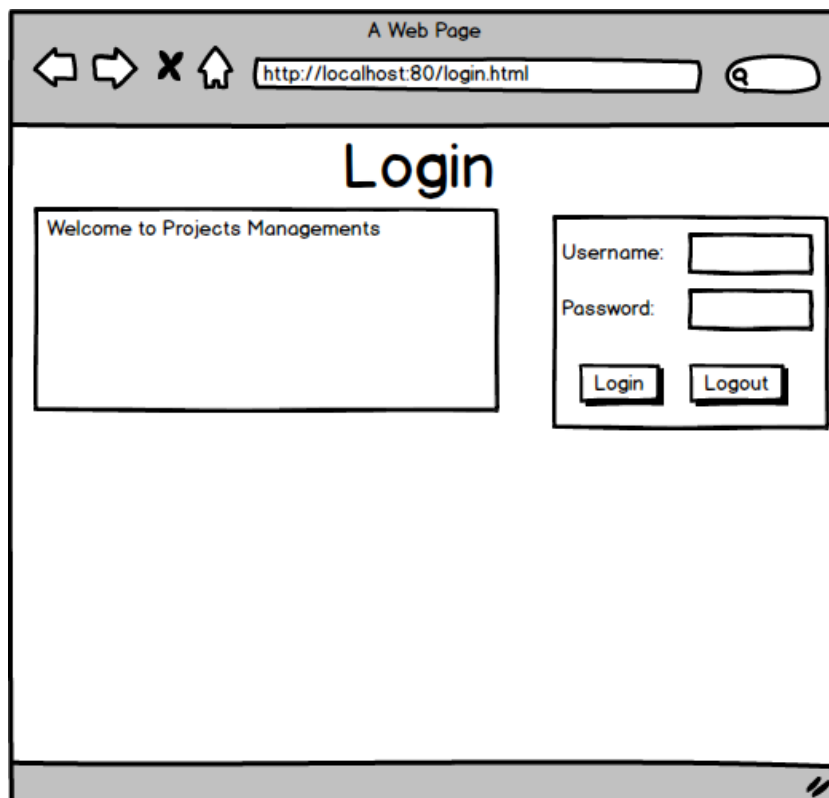
Porpose new End date:  📅

Vote:  ▼

Justification:

Figura 67 - Team Member – Votação.

### A.3 STORYBOARD DO STAKEHOLDER



A Web Page

Navigation icons: back, forward, close, home

Address bar: <http://localhost:80/login.html>

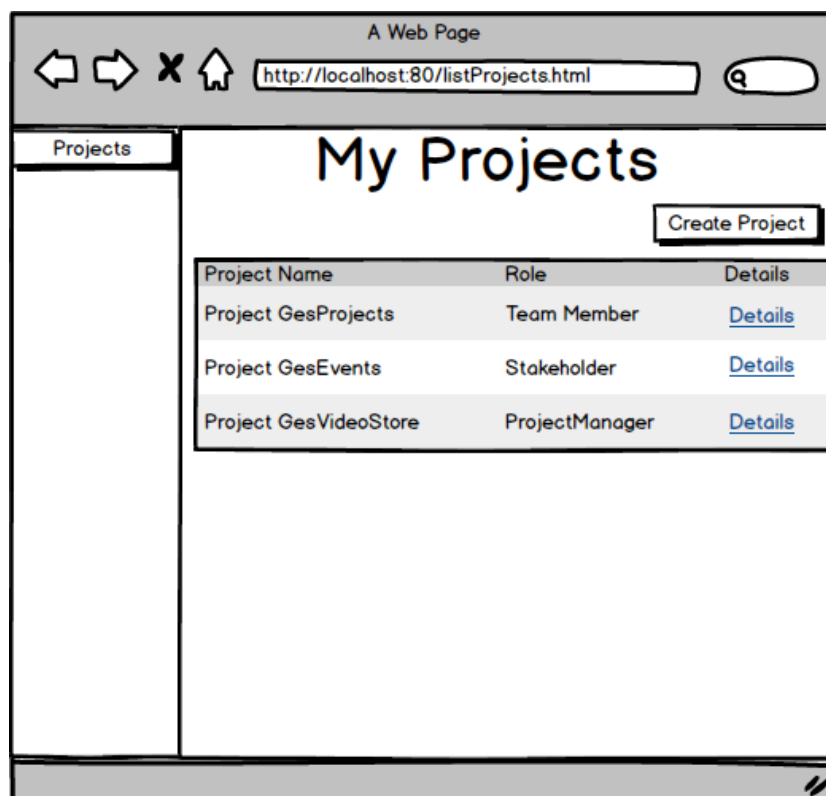
# Login

Welcome to Projects Managements

Username:

Password:

Figura 68 - Stakeholder - Login.



A Web Page

Navigation icons: back, forward, close, home

Address bar: <http://localhost:80/listProjects.html>

# My Projects

Project Name	Role	Details
Project GesProjects	Team Member	<a href="#">Details</a>
Project GesEvents	Stakeholder	<a href="#">Details</a>
Project GesVideoStore	ProjectManager	<a href="#">Details</a>

Projects

It should select the project that we want to work.

Figura 69 - Stakeholder - Meus projetos.

A Web Page

http://localhost:80/project.html

Projects

Requirements

Change request

Votation change request

## Project

Name: Project GesProjects

Description: Project to managem

Start Date: 2015/02/01

End Date: 2015/12/01

Save Cancel

List of all fields the selected project.

Figura 70 - Stakeholder - Projeto.

A Web Page

http://localhost:80/listRequirement.html

Projects

Requirements

Change request

Votation change request

## List of Requirements

Create requirement

Requirements	Date creation	
requirement A	2015/03/03	<a href="#">Details</a> <a href="#">Remove</a>
requirement B	2015/05/05	<a href="#">Details</a> <a href="#">Remove</a>

Figura 71 - Lista de requisitos.

A Web Page

http://localhost:80/listRequirement.html

Projects

Requirements

Change request

Votation change request

# Requirements

Name: requirement B

Description: description of requirement B

Status: created ▼

Degree importance: 1

Save Cancel

Figura 72 - Stakeholder - Requisito.

A Web Page

http://localhost:80/listChange.html

Projects

Requirements

Change request

Votation change request

# List of change request

Create change

Description	Date	Status
change request A	2015/03/03	asked <a href="#">Edit</a> <a href="#">Remove</a>
change request B	2015/05/05	votation <a href="#">Edit</a> <a href="#">Remove</a>

Change request list that we are working. Link Edit allows to going edit change request selected.

Figura 73 - Stakeholder - Lista de pedidos de alteração.

A Web Page

http://localhost:80/changeRequest.html

Projects

Requirements

Change request

Votation change request

## Change Request

Description: change to description of project or start and end date

Start date: 2015/06/05

End date: 2015/11/06

Status: asked

Save Cancel

Figura 74 - Stakeholder - Pedido de alteração.

A Web Page

http://localhost:80/listVotationRequest.html

Projects

Requirements

Change request

Votation change request

## List of votation

Description	Date	Status	
change request B	2015/05/05	votation	<a href="#">Vote</a>

The option Votation change request is enable if stakeholder belongs Analysis Team.

Votation request list that we are working. Link Vote allows to going vote on the change request selected.

Figura 75 - Stakeholder - Lista de votações.

A Web Page

← → × 🏠  🔍

Projects

Requirements

Change request

Votation change request

## Votation Request

Description:

Requirement:  ▼

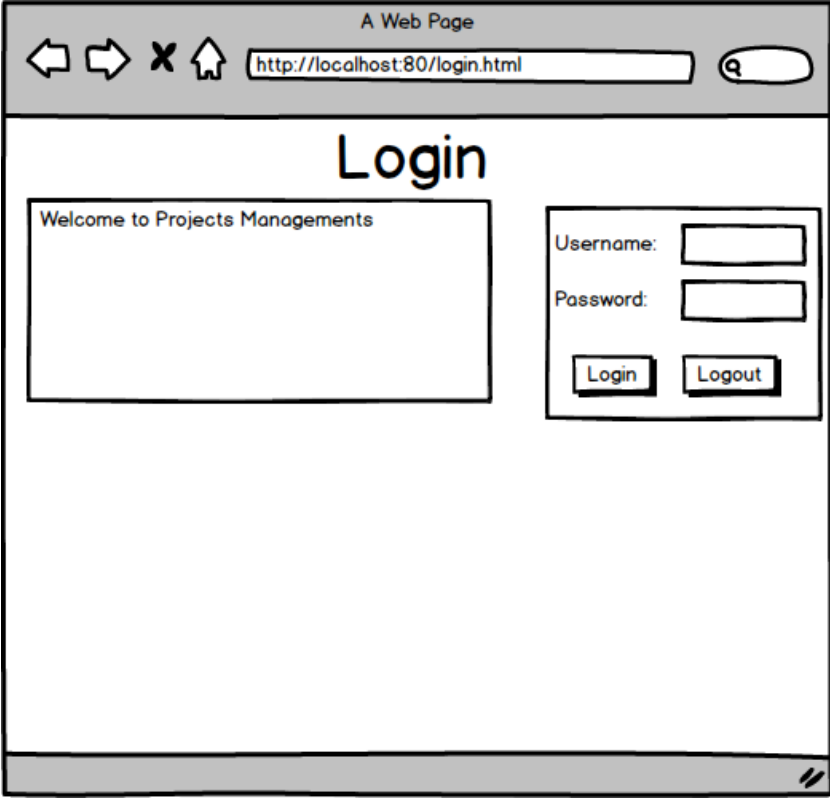
Porpose new End date:  📅

Vote:  ▼

Justification:

Figura 76 - : Stakeholder - Votação.

#### A.4 STORYBOARD DO ADMINISTRADOR



A Web Page

http://localhost:80/login.html

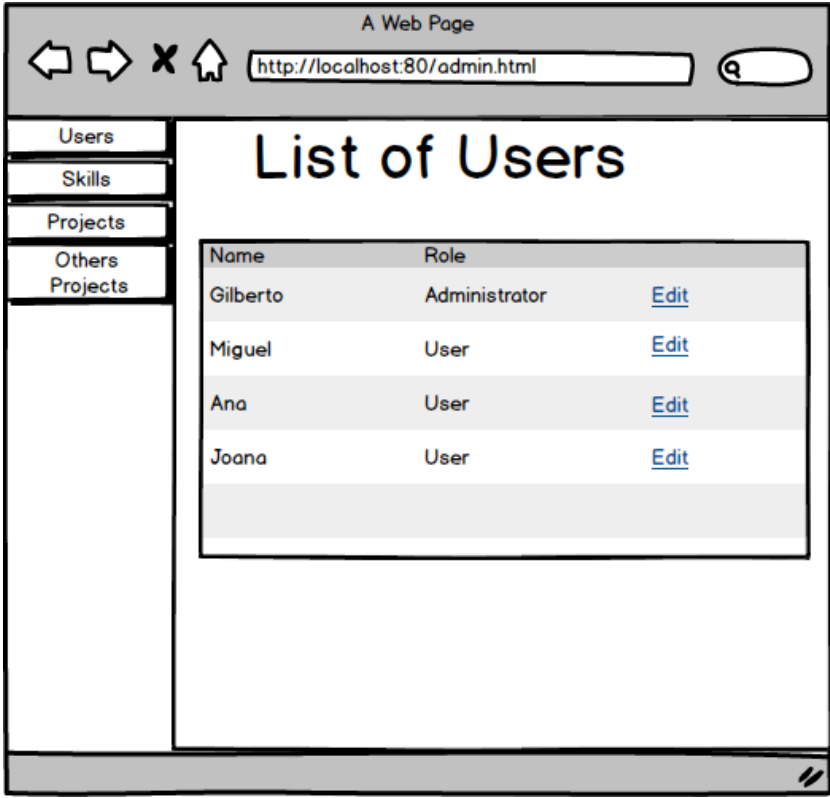
# Login

Welcome to Projects Managements

Username:

Password:

Figura 77 - Administrator - Login.



A Web Page

http://localhost:80/admin.html

- Users
- Skills
- Projects
- Others Projects

# List of Users

Name	Role	
Gilberto	Administrator	<a href="#">Edit</a>
Miguel	User	<a href="#">Edit</a>
Ana	User	<a href="#">Edit</a>
Joana	User	<a href="#">Edit</a>

Figura 78 - Administrator - Lista de utilizadores.

A Web Page

http://localhost:80/user.html

Users

Skills

Projects

Others

Projects

## User

Name:

Username:

Password:

Email:

☐ Active User

☐ Administrator User

Save Cancel

Figura 79 - Administrator - Utilizador.

A Web Page

http://localhost:80/listSkill.html

Users

Skills

Projects

Others

Projects

## List of Skills

Create Skill

Skill	Details
Analist	<a href="#">Edit</a> <a href="#">Remove</a>
Programmer	<a href="#">Edit</a> <a href="#">Remove</a>

Figura 80 - Administrator - Lista de habilidades.



A Web Page

http://localhost:80/listSkill.html

Users  
Skills  
Projects  
Others Projects

## Skill

Name:

Save Cancel

Figura 81 - Administrator - Habilidade.

A Web Page

http://localhost:80/listProjects.html

Users  
Skills  
Projects  
Others Projects

## My Projects

Change Project Manager Create Project

Project Name	Role	Details
Project GesProjects	Team Member	<a href="#">Details</a>
Project GesEvents	Stakeholder	<a href="#">Details</a>
Project GesVideoStore	ProjectManager	<a href="#">Details</a>

It should select the project that we want to work

Figura 82 - Administrator - Meus projetos.



A Web Page

http://localhost:80/editProject.html

Users

Skills

Projects

Others

Projects

## Project

Name: Project GesProjectsSeaPorts

Description: Project to seaports management

Start Date: 2015/08/01

End Date: 2015/08/24

Project Manager: Ana ▼

Save Cancel

List of all fields the selected project.

The project manager is enabled for changing.

Figura 85 - Administrator - Outro projeto.

## ANEXO B: SCREENS DA APLICAÇÃO

### B.1 SCREENS DO ADMINISTRADOR

The screenshot displays the Administrator interface. At the top, a dark grey header bar is visible. Below it, a light grey box contains a welcome message 'Welcome, Miguel!' and a 'Sign Out' button. The main content area is divided into two sections. On the left, the 'Edit User' form is shown, featuring fields for Name, Username, Password, and Email, each with a 'Required' label. The Name field contains 'Andre', the Username field contains 'AndreAndrade', the Password field is masked with dots, and the Email field contains 'andreandrade@gmail.com'. Below these fields are two checkboxes: 'User active' (checked) and 'Admin Role' (unchecked). At the bottom of the form are 'save' and 'cancel' buttons. On the right, the 'Administrator Operations' sidebar contains four blue buttons: 'Users', 'Skills', 'Projects', and 'Others Projects'.

Welcome, Miguel!

Sign Out

## Edit User

Name: Required  
Andre

Username: Required  
AndreAndrade

Password: Required  
.....

Email: Required  
andreandrade@gmail.com

☒ User active ☐ Admin Role

save cancel

### Administrator Operations

Users

Skills

Projects

Others Projects

Figura 86 - Adminitrator - Screen do utilizador.

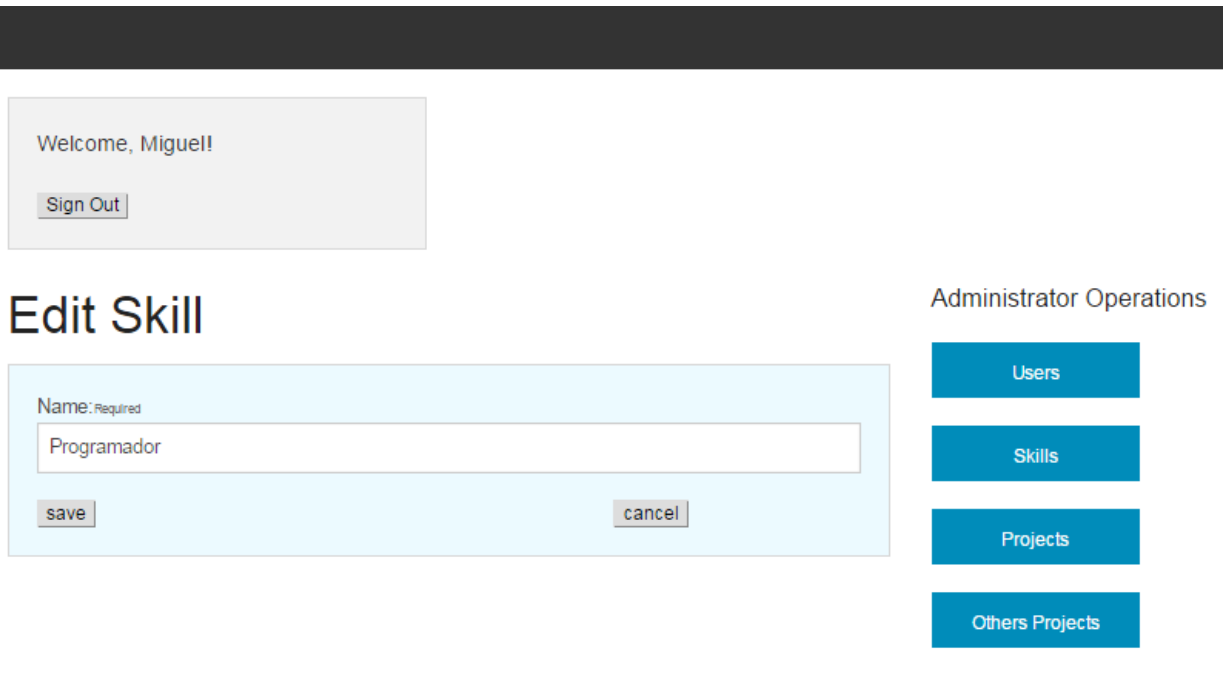


Figura 87 - Administrator - Screen da habilidade

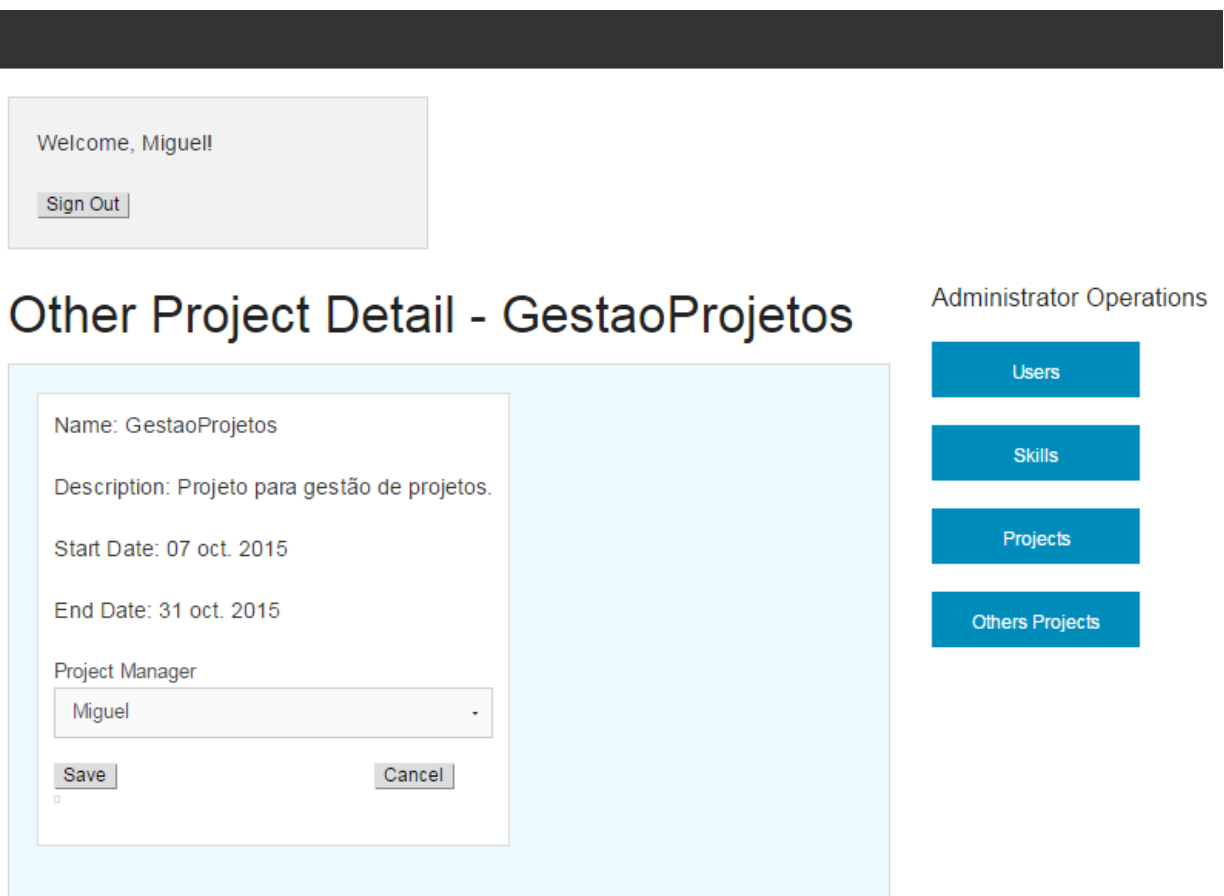


Figura 88 - Administrator - Screen da mudança do chefe de projeto.

## B.2 SCREENS DO PROJECT MANAGER

Welcome, Miguel!  
[Sign Out](#)

### Edit Project - GestaoProjetos

#### Team Member

User:  
andré

[\[add\]](#)  
[\[remove\]](#)

Team Members  
Associated  
Gilberto  
Miguel

Skill:  
Programador  
Analista

management skill:  
[\[add skill\]](#)  
[\[remove skill\]](#)

Skill Associated:  
Programador

[back](#)

Administrator Operations

[Users](#)

[Skills](#)

[Projects](#)

[Other Projects](#)

Try one of these buttons:

Project Manager Operations

[Team Members](#)

[Stakeholders](#)

[Requirements](#)

[Analysis Team](#)

[Use Cases](#)

[Tasks](#)

[Changes](#)

[Validation](#)

Figura 89 - Project Manager - Associar membro de equipa.

Welcome, Miguel!  
[Sign Out](#)

## Edit Project - GestaoProjetos Stakeholder

User:

Role:

Stakeholder Associated  

andré

[\[remove\]](#)  
[\[edit\]](#)

Role defined:  

Administrador

[back](#)

### Administrator Operations

[Users](#)

[Skills](#)

[Projects](#)

[Others Projects](#)

Try one of these buttons:

#### Project Manager Operations

[Team Members](#)

[Stakeholders](#)

[Requirements](#)

[Analysis Team](#)

[Use Cases](#)

[Tasks](#)

[Changes](#)

[Votation](#)

Figura 90 - : Project Manager - Associar Stakeholder.

Welcome, Miguel!

Sign Out

## Edit Project - GestaoProjetos

### Analysis Team

Team Members:

[add Team Member]  
[remove Team Member]

Team Members Associated:

Miguel  
Gilberto

Stakeholders:

[add stakeholder]  
[remove stakeholder]

Stakeholders Associated:

andre

back

Administrator Operations

Users

Skills

Projects

Others Projects

Try one of these buttons:

Project Manager Operations

Team Members

Stakeholders

Requirements

Analysis Team

Use Cases

Tasks

Changes

Violation

Figura 91 - Project Manager - Equipe de análise.



